Processing Archive Information in Digital University

Alexander V. Baldin[1*] and *Dmitriy V.* Eliseev [1]

[1]Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. The article discusses methods of processing and storing data archive used in the digital university. Disadvantages of these methods are found. As a result, a fundamentally new method of processing and storing information archive in a constantly changing scheme database is proposed. This method uses mivar technologies. The multidimensional space structure has been developed to store the data archive. This multidimensional space describes the temporal relational model. For processing data, archive is proposed scheme for selecting the subspace and converting it into relations. A method of transformation of relational databases into multidimensional mivar space for efficient execution of operations on temporal data with changing structure is proposed. The transition to a multidimensional space allows us to describe the process of changing temporal data and their structure in a unified way. As a result, the time required to adapt the database schema and the redundancy of information storage are reduced. The results of this work are used in the human resource management database of BMSTU.

1 Introduction

The digital university is a complex, heterogeneous and constantly changing system that must consolidate and integrate a huge number of subsystems and modules into a single information space [1]. Most of the data stored in the digital university subsystems are temporal, i.e. they are relevant for a certain period of time [2,3,4]. Such data include passport data, the state of education and educational program of students, information about teachers and their individual plans, etc. During the functioning and maintenance of the digital university, an archive is accumulated: new data is constantly entered into the system, and old data becomes out of date, but is not deleted from the system. Because legislation and requirements to subsystems of digital university are constantly changing, Against the background of constantly changing legislation and requirements to the TS subsystems, it takes place the modernization of already implemented business processes and databases [5,6,7].

Currently, the relational model and its modifications is the most popular and widespread in databases [8,9,10]. The known temporal data models that extend the relational model and considered in [11,12,13,14,15,16] have a number of disadvantages that

* Corresponding author: bal@bmstu.ru

constrain their application in practice. The use of these models in systems that need to change the data structure over time leads to an increase in redundancy of information storage, as well as to basic relational model growth. It complicates the compilation, execution of queries and adaptation of the database to new tasks.
The process of adapting the database with temporal data is shown in fig. 1.

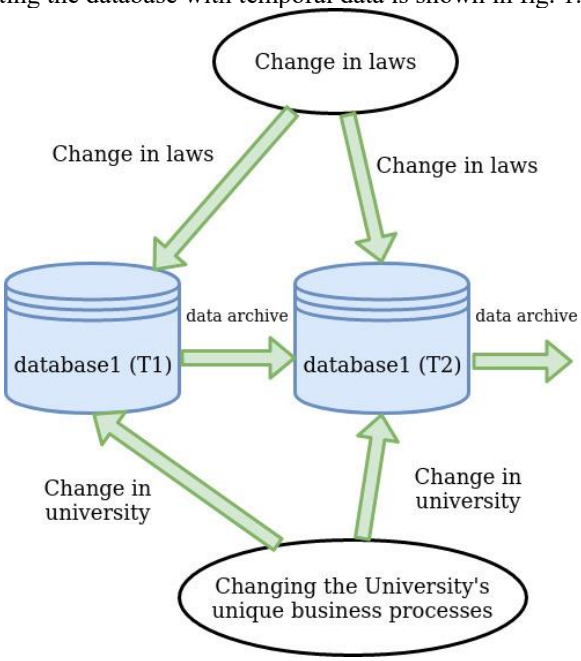

Fig. 1. The database change scheme of a digital university subsystem.

Currently, the implementation of work with the data archive accumulated in previous systems can be performed in 2 ways:

1. Transfer of archive from several databases (DB) to one, the last. For fig. 1 data from database1 at time T1 is moved to database1 at time T2, when there was a change in the database schema. This approach leads in data integrity damaging because database schemas different. Usually data transfer has long implementation times.
2. To increase the number of databases in the system or increase the number of relations in the database schema when it is changed. This complicates the compilation and execution of queries to temporal data, because they have to be joined from different databases or from different relations.

Currently, there is a new mivar approach [17,18,19] to the description of changeable, dynamic subject areas, which allows to fix data storage redundancy. The mivar approach is used for a class of learning systems whose task is to study and model complex dynamic subject areas. The basis of the mivar approach to data presentation is an integral, unified description of subject area from different points of view through a multidimensional space. However, the use of this approach to work with temporal relational database is impossible, as it is necessary to implement a mechanism for processing multidimensional representation of changing relations [20].

The use of mivar technologies for storage and processing of information archive allows fixing the above disadvantages. The development of a multidimensional model of storage and processing of temporal data archive is an actual task. The use of mivar technologies will increase the system life cycle by adapting the database of the information system functioning in a dynamically changing subject area with temporal data.

2 Storage of temporal data with variable structure

Consider a relation with temporal tuples. In this case, the relation consists of tuples, which define the states of the domain objects. The dependency graph of the size of a relation with temporal tuples is shown in fig. 2.

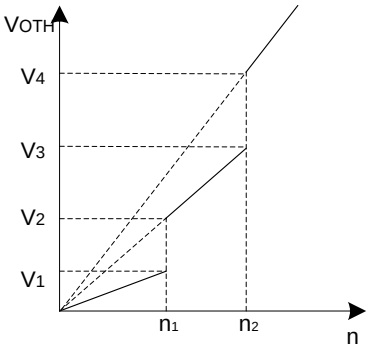

Fig. 2. Dependence of the size of a relation with the changing scheme from the number of temporal tuples.

In the graph, the *n* axis is the number of tuples in the relation. If the relation schema is unchanged, the size of the relation grows linearly by the tuples number increase, $V_{OTH} = kn$, where *k* - number of attributes in the relation schema. In this case changing the relation schema consists only in adding new attributes to keep the previously accumulated history. Therefore, the relation size increases incrementally, as space for new attributes is allocated in existing tuples. The change in the relation scheme is shown in the graph when the number of tuples is $n=n_1$ and $n=n_2$. If you change the relation scheme, when $n=n_1$, the relation size increases from V_1 to V_2, and the graph angle changes. Similarly happens when $n=n_2$.

Mathematically, the dependence of the size of a relation with a changing scheme on the number of temporal tuples is presented in formula 1.

$$V_{OTH} = \begin{cases} kn,\ 0 \le n < n_1, \\ (k+\Delta k_1)n,\ n_1 \le n < n_2, \\ (k+\Delta k_1+\Delta k_2)n,\ n_2 \le n < n_3, \\ \ldots \end{cases} \quad (1)$$

After each change in the relation scheme, the line inclination angle increases, i.e. the relation size increases faster. Because attributes are not removed from the relation schema, over time the relation contains a large number of unused, old attributes. It results in inefficient storage of information.

3 The structure of multidimensional information space for storing data archive accumulated in previous versions of the system

"Mivar" space (Multidimensional Information Varying space) is a self - organizing dynamic multidimensional object-system discrete space of unified data representation and rules [20].

Consider the structure of the mivar space for consolidation of relational databases. A relational data model is a set of normalized relations to which relational algebra operations apply. Each relation includes many attributes and many records that are defined by the relation key. Thus, to describe a relational data model in a mivar space, you must enter three axes: the relation axis, the relation attribute axis, and the relation record identifier axis. A time axis is added that determines the relational model state.

Thus, the structure of the mivar space for consolidation of relational databases consists of four main axes:

V – the set of relations of relational model.

$V = \{v_i\},\ i = \overline{1, I_V},\ I_V = |V|.$

S – the set of relations attributes .

$S = \{s_i\},\ i = \overline{1, I_S},\ I_S = |S|.$

ID – the set of relations record identifiers.

T – the set of state change times of a relational database.

Then the multidimensional space will have the following:

$$M = V \times S \times ID \times T.$$

If $m \in M$, then *m=(v, s, id. t)* – point of multidimensional space.

The relation, attribute, record ID subspace defines the state of the relational model that depends on the other axes. For each axis, a set of elements from the original relational model is generated. The Cartesian product of these sets generates a multidimensional space for a temporal relational data model

In multidimensional space, each tuple attribute value of a relation matches to a point with certain coordinates. The point matching to the selected attribute value is shown on fig. 3. The set of all points in a multidimensional space matches to a relational data model. In the mivar approach, space points that store the corresponding attribute values of the relational model define the data structure. Thus, the data structure is determined by the data that is stored in the mivar space.

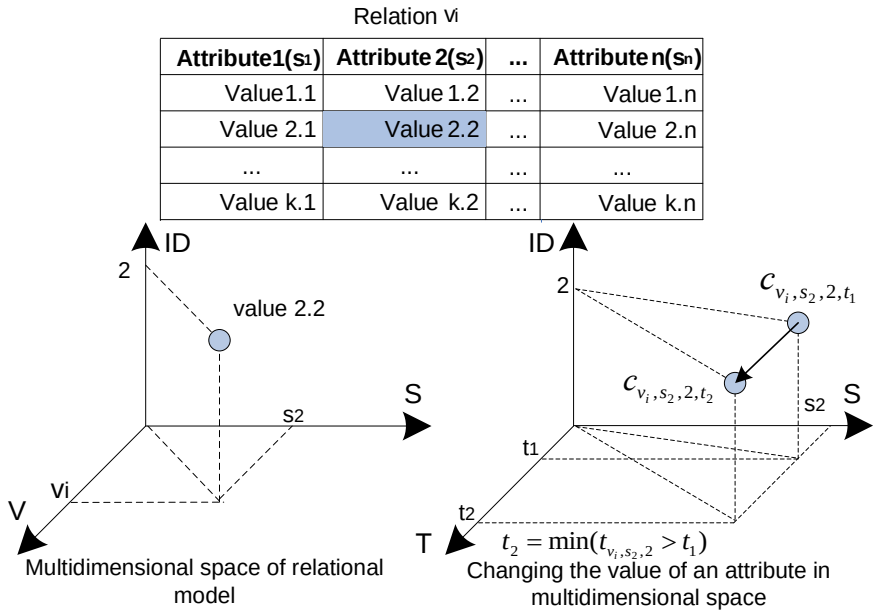

Fig. 3. Relational and multidimensional data representation.

The structure of relation vi and the mivar description of this relation are presented on fig. 2. When designing a relational model, the sets that describe the structure of the model: relations, attributes, tuples, are dependent on each other. When you destroy some element of the set of the higher level, all dependent sets are destroyed. When constructing a multidimensional data representation space, all sets are independent of each other that creates new opportunities to change the database schema in the relational model and preconditions for the development of new methods of data processing with a variable structure.

In relational databases, the uniformity of filling and storing the data of all relation attributes leads to the fact that if no values are specified, then the database still stores a record (empty) of fixed length. For many domains, this leads to an unjustified waste of computational resources. In the multidimensional space, only the necessary attribute values are stored.

Thus, the mivar representation of the relation in comparison with the relational one is more general, having more opportunities for representation, change of both data and their structure. The mivar space allows you to save data from previous databases without changing their structure that provides the use of existing SQL queries.

4 Transform a relational database into the multidimensional space

The transformation of the original relational model into the multidimensional space is performed to produce the initial state of the multidimensional space presented in section 3. To perform this operation, we introduce a transformation operator for a relational data model into the multidimensional space describing a temporal relational model: $\varphi: R_D \to C_M$. The following actions are performed:

1. $V = \{vr_1, vr_2, ..., vr_K\}$, where vr_i – relation name r_i, $i = \overline{1, K}$.
2. $S = \bigcup_{i=1}^{K} R_i$.
3. Dr_i – a set, consequently there is a one-to-one matching $f: Dr_i \to N_{Dr_i}$, where $N_{Dr_i} = \{1, 2, ..., |Dr_i|\}$, - cardinality of Dr_i. Then $ID = \bigcup_{i=1}^{K} N_{Dr_i}$.
4. $T=\{t_0\}$.
5. $c_{vr_i, R_i A_j, n, t_0} = Dr_{in}.R_i A_j$, $i = \overline{1, K}$, $n = \overline{1, |Dr_i|}$, $j = \overline{1, Pi}$, где $Dr_{in}.R_i A_j$- attribute value R_iA_j in tuple n of relation r_i, P_i – number of attributes in relation scheme R_i.

The transition from a relational data model to a multidimensional space with the help of the introduced transformation φ allows us to describe the process of changing the model. Model change occurs by adding new points in multidimensional space, and the introduced changes do not affect the original relational model. The coordinates of the points define the data structure. As a result, changing the data structure and changing the data in relations are performed simultaneously in the multidimensional space. This representation of the relational model allows you to keep a changes history for each attribute in relations separately, that minimizes the total number of relations in the temporal relational model.

5 The inverse points values transform from multidimensional space to relational data model

The transformation operator α receive from certain set of point values in a multidimensional space the relevant state of the relational data model: $\alpha : C_M \to R_D$. The following actions are performed:

1. First, sets of point values C_{r_i}, describing relations, contained in the based set C_M are defined. V_{C_M} - the set of relations names contained in the based set C_M.

2. For each set of point values C_{r_i}, matched to the relation r_i, the relation scheme is defined, which consists of many attributes $S_{C_{r_i}}$, contained in the set C_{r_i}, and record id.

3. The attribute values of relation tuples are generated Dr_{in} from point values of a multidimensional space with corresponding coordinates.

4. The relational data model is generated from received schemes and sets of relations tuples R_D.

Formally α - transformation is described as follows:

1. $C_{r_i} = \{c_m : v = vr_i,\ c_m \in C_M\},\ i = \overline{1, |V_{C_M}|}$.

2. $R_i = S_{C_{r_i}} \cup \{id\},\ i = \overline{1, |V_{C_M}|}$.

3. $Dr_{in} = \{c_m : id = id_n,\ c_m \in C_{r_i},\ c_m \neq null\},\ (\forall Dr_{in} \neq \varnothing) \Rightarrow Dr_{in} = Dr_{in} \cup \{id_n\},$
$Dr_i = \{Dr_{in}\},\ i = \overline{1, |V_{C_M}|},\ ID_{C_{r_i}} = \{id_n\},\ n = \overline{1, |ID_{C_{r_i}}|}$.

4. $r_i = (R_i, Dr_i),\ R_D = \{r_i\},\ i = \overline{1, |V_{C_M}|}$.

Thus, with the help of the introduced α-transformation from the multidimensional representation of the temporal relational model, it is possible to get certain relational model states for their next processing by standard SQL queries. This allows you to apply existing queries in the information system to the relational database when moving it to a multidimensional space.

6 Work with multidimensional space for processing the archive information

Working with a multidimensional space for processing information archive consists of 3 steps.

1 step. Transforming the original relational data model into a multidimensional space.

The first step in processing the information archive is the transformation of the source database into a multidimensional representation, proposed in section 3. The source database is broken down into sets of elements (relationships, attributes, attribute values, and record identifiers). Based on these set elements, a multidimensional representation of relational database is generated.

2 step. Changing the domain model in multidimensional space.

The change of the relational model is performed in the multidimensional space. The process of changing the relational model occurs by adding new points to the multidimensional space. The coordinates of points define the data structure

Changing the multidimensional space can be done 2 types

1."Analysis and formalization of changes in objects of the subject area". The result of this function is a change in the multidimensional space: either the structure of the space

changes (new axes are added), or the sets that define the axes of the space change (new elements are added to these sets).

2."Entering new data and changing data". The result of this function is to store values of new points in multidimensional space.

The process of changing the domain model is iterative (fig. 4), i.e. each next state of the domain model depends on the previous one: $C_i=\delta(C_{i-1})$, where δ – function that specifies the change in the model current state from the previous one (the set of points values with coordinates that appeared in space at time t_i).

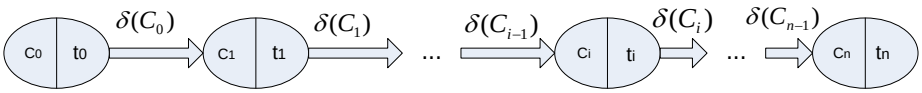

Fig. 4. The process of changing the domain model.

In addition, this process takes place in time, i.e. each state of the domain model is valid in a certain period of time: C_i, $t \in [t_i, t_{i+1})$. The transformation of a relational data model into a multidimensional space allows us to describe and store the process of changing the model over time.

3 step. Query execution to the multidimensional space.

This step consists of three subitems

3.1. Subspace selection that contains the query result. At this subitem, operations are performed on the coordinates of points in multidimensional space, without analyzing the values of these points. These are operations on selection of the various subspaces corresponding to certain conditions on axes (operations getting slices of multidimensional space). As a result, a subspace containing the query result is selected from the all multidimensional space.

3.2. Transform a selected subspace into a relational data model. To apply existing queries to the database, it is necessary to convert the selected multidimensional representation of temporal data back to relational, fig. 5.

3.3. Getting the result of the query. The result of the query is calculated from the selected relational data model using relational algebra operations (SQL language).

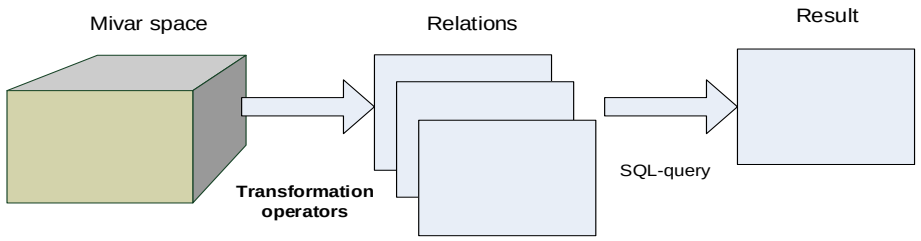

Fig. 5. The work scheme with multidimensional information space.

Operations with point coordinates select the required subspace. Then the selected subspace is transformed into domain relations. Relational algebra operations apply to obtained relations to determine the result of the query (fig. 5).

Existing SQL queries to relational databases consolidated within the mivar space and containing data archive from previous systems are used with additional perators that convert a specific area of the mivar space to the relevant relations state. Thus, processing mivar representation of the data archive includes a single initial transformation from the

relational data model to the multidimensional space and then to work with this multidimensional model: the input of new data, modification of data structures and query execution to multidimensional representation of temporal relational model with data archive.

7 Adaptation time estimation of the information system

Adaptation of the information system is performed when the database schema is changed. For a relational database, adaptation includes the following steps:

1. Create a new empty DB (t_{CDB}).
2. Creating a new modified database scheme (t_{MDB}).
3. Data transfer to a new database ($t_{TRANSFER}$).
4. Application modification (t_{APP}).

Thus, the adaptation time of the information system with a relational database depends on four variables:

$T_{AIS}=f_P(t_{CDB}, t_{MDB}, t_{TRANSFER}, t_{APP})$.

Usage of the technique proposed in section 6 allows to exclude steps 1-3 at adaptation of information system. Thus, the adaptation time of the information system by the use of the proposed technique depends on the application modification time:

$t_{AIS}=f_M(t_{APP})$.

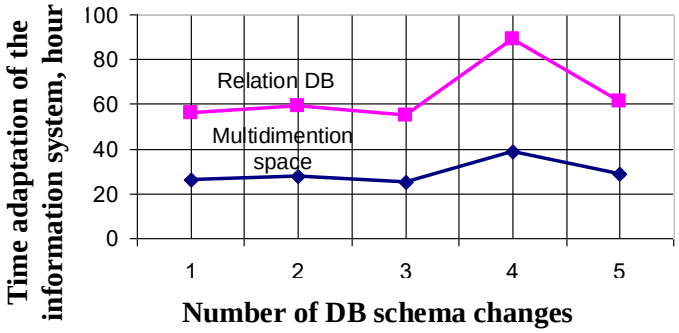

Fig. 6. Comparing adaptation time of the information system with a relational database and a multidimensional space.

Experiments were performed for five DB scheme changes that were in practice with the University's personnel system. For fig. 6 database schema changes were performed when solving the following tasks:

1. Dynamic creation of the University's structure on a given date.
2. Development of the competition functional block for teachers.
3. Change of work with financing sources.
4. The transformation from a basic rate to a floating salary.
5. Accounting in personnel system teachers who work with disabled students.

The peak on the chart is associated with the fourth task. It happened because the transformation from a basic rate to a floating salary led to the modernization of all key relations, screen forms and reports of the information system.

Performed researches have shown that the average adaptation time of the information system when using the developed technique reduced by 53%.

8 Conclusion

In the article, structure of the space for temporal relational data model was discussed. It stores archive information and consists of 4 main axes: axis of relations, axis of attributes, axis of tuple identifiers, and the time axis. To work with such a multidimensional space, you need to select a part of the space, convert it into relations, to which you can perform SQL queries and get the necessary data.

As a result, it became necessary to extend the standard SQL statements with new ones that allow selecting the required parts of a multidimensional space and transforming them into the relevant relations states

The proposed approach for processing archive data in a continuously changing their structure allows to transfer information from one database to another and have relevant and non-redundant database schema, that simplifies storage and handling of archive information in the digital university.

References

1. V.M. Chernenkiy, A.V. Baldin, S.A. Tonoyan, *BMSTU information management system "Electronic University."*, Ed. BMSTU 2009, 376, pp. 304-325 (2009)
2. S.A. Tonoyan, V.B. Timofeev, S.V. Chernenkiy, *"Analysis and configuration network of financial-economic activity of BMSTU on a platform of "1C: Enterprise 8. '" BMSTU bulletin*, a series of "Instrument", Special issue **№5** pp. 101 - 108
3. R.T. Snodgrass, I. Ahn, *Temporal databases*, IEEE computer, **19(9)**, pp. 35-42 (1986)
4. A.V. Baldin, D.V. Eliseev, K.G. Agayan, *Review of methods for constructing temporal systems based on relational databases*, Science and Education (BMSTU), **№7**, July 2012
5. D.V. Eliseev, *Methods of processing temporal relational database in MIVAR space: dis ... Ph.D.*, M. BMSTU, Moscow, p. 149 (2011)
6. A.V. Baldin, S.A. Tonoyan, D.V. Eliseev, Personnel data archive processing in mivar space by means of 1C, Engineering Journal: Sciense and Innovation, **№ 11 (23)**, p. 7 (2013)
7. S.A. Tonoyan, A.V. Baldin, D.V. Eliseev, *Methods of upgrading the standard modules of typical configuration based on the technological platform 1C: Enterprise 8 "with minimal modifications."*, Science and Education (BMSTU), №8, August 2012 (2012)
8. Y.A. Grigoriev, *Synthesis algorithm partially optimized relational database schema*, Science and Education (BMSTU), №1
9. V.M. Chernenkiy, Yu.E. Gapanyuk, A.A. Mavzyutov, Development of complex biomedical information systems based on adaptive objects, BMSTU Bulletin. Series "Instrument", Special issue 2011 **№ 3** "Biometric technology", pp. 105 – 112 (2011)
10. M.V. Vinogradova, E.G. Igushev, *Database Designer based on entities, and their details with the normalization possibility*, Science and Education (BMSTU), № 01 (2012)
11. A.A. Tansel, *Generalized relational framework for modeling temporal data*, Temporal databases: theory, design, and implementation, A. Tansel, J. Clifford, S. Gadia et al. – Benjamin, Cummings Publishing Company, pp. 183-201 (1993)

12. T. Tsuji, A. Hara, K. Higuchi, An extendible multidimensional array system for MOLAP, Proceedings of the 2006 ACM symposium on applied computing, pp. 503-510 (2006)
13. K. Torp, R.T. Snodgrass, C.S. Jensen, *Modification semantics in now-relative databases*, Information systems, **29(8)**, pp. 653–683 (2004)
14. K. Torp, R.T. Snodgrass, C.S. Jensen, *Effective timestamping in databases*, The VLDB journal, **8(4)**, pp. 267-288 (2000)
15. E. McKenzie, R.T. Snodgrass, Schema evolution and the relational algebra, Information systems, **15(2)**, pp. 207–232 (1990)
16. C.S. Jensen, R.T. Snodgrass, M.D. Soo, Unifying temporal data models via a conceptual model, Information systems, **19(7)**, pp. 513-547 (1994)
17. O.O. Varlamov, *Evolutionary knowledge and information base for the adaptive synthesis of intelligent systems. MIVAR information space*, Radio and communication, Moscow, p. 286 (2002)
18. O.O. Varlamov, D.A. Chuvikov, D.V. Aladin, L.E. Adamova and V.G. Osipov, *Logical artificial intelligence Mivar technologies for autonomous road vehicles*, IOP Conference Series: Materials Science and Engineering, **534(1)**, 012015 (2019)
19. D.A. Chuvikov, O.O. Varlamov, D.V. Aladin, V.M. Chernenkiy and A.V. Baldin, *Mivar models of reconstruction and expertise of emergency events of road accidents*, IOP Conference Series: Materials Science and Engineering, **534(1)**, 012007 (2019)
20. A.V. Baldin, S.A. Tonoyan, D.V. Eliseev, *Query language to mivar representation of relational databases, containing information archive from previous human resources systems*, Engineering Journal: Sciense and Innovation, **№ 11 (23)**, p. 20 (2013)