Algebraic Invariants of Database Schemes

Evgeny Beniaminov*
Russian State University for Humanities,
Chajanov 15, Moscow, RUSSIA
email: beniamin@infao.rggu.msk.su

Abstract

In this paper, following [10, 11] we consider some applications of category theory to the modeling of database systems. In particular, we translate of the notions of join dependencies [3] to the notions of homology algebra [8]. This approach gives a new possibility for the research of database systems.

1 Introduction and motivation

Todays many application need access to information stored and distributed among multiple databases. One of solutions of this problem is a design of new global database systems (or global views) from local existing databases. Schemes of such global databases usually are very large and we need instruments to solve next problems at the creation of large schemes from schemes of local databases:

1. Let Sch and Sch' be two schemes of databases. Are Sch and Sch' equivalent?
2. Let Sch and Sch' be two schemes of databases. Does there exist an interpretation from the scheme Sch' to the scheme Sch?

[0]The research was supported by the Russian Basic Research Foundation, grant number 94-01-01479

3. Suppose Sch and Sch' are schemes of databases, Sch'' is a subscheme of Sch', and $g : Sch'' \to Sch$ is an interpretation. Does there exist an extension of g to $Sch' \supset Sch''$?

The schemes Sch and Sch' are equivalent if there exist interpretations $i : Sch \to Sch'$ and $j : Sch' \to Sch$ such that the compositions $j \circ i : Sch \to Sch$ and $i \circ j : Sch' \to Sch'$ are equivalent to the identity interpretations.

Let Sch be a scheme of a database. Denote by $\mathcal{S}(Sch)$ the set of states of the database with the database scheme Sch. If $f : Sch' \to Sch$ is an interpretation from the local database scheme Sch' to the global database scheme Sch, then there exists the functions $f^* : \mathcal{S}(Sch) \to \mathcal{S}(Sch')$. The function f^* takes the state $\mathcal{S} \in \mathcal{S}(Sch)$ of the global database to the state $\mathcal{S}' = f^*(\mathcal{S}) \in \mathcal{S}(Sch')$ of the local database.

Now we discuss notions of algebraic invariants of database schemes and what kind of help can be from the algebraic invariants for to arrive at solutions of the problems 1–3.

An algebraic invariant I of database schemes is a functor from the schemes category to algebras category. If Sch is the scheme, then $I(Sch)$ is the invariant. If the schemes Sch and Sch' are equivalent, then the algebras $I(Sch)$ and $I(Sch')$ isomorphic. If there exists an interpretation $f : Sch' \to Sch$, then there exists a homomorphism $I(f) : I(Sch') \to I(Sch)$. That is why if there doesn't exist an extension of the homomorphism $I(g) : I(Sch'') \to I(Sch)$ to $I(Sch') \to I(Sch)$, then there doesn't exist an extension of g to $Sch' \supset Sch''$.

In this paper, I suggest to consider a cohomology of databases schemes as the invariant. Cohomologies were applied in topology for the investigation of complex topological spaces [6]. Then cohomologies were defined and applied for more general case in topos theory [8]. These definitions can be used for databases schemes when the databases schemes are systems of join dependencies or, in general, are represented by a many-sorted first-order predicate language $\mathbf{L}$ with equality, with the logical constants TRUE, FALSE, with the logical symbols $\wedge$,$\vee$,$\exists$ and without the symbols $\neg$, $\to$, and $\forall$. Such language $\mathbf{L}$ is called a geometric language(see [8]). Suppose also that integrity constraints of databases are a set of sequents $(IF\varphi, THEN\psi)$, where φ, ψ are formulas of the geometric language $\mathbf{L}$. Thus we suppose that the scheme Sch is defined by a finite geometric theory $\mathbf{T} = (\mathbf{L}, \mathbf{I})$, where **I** is a finite set of sequents. Examples of such schemes are schemes with sets of functional,joint, key, inclusion, Horn dependencies.

I try show that many problems of theory database may be translate to well-known problems category theory and cohomology theory.

This paper is organized as follows. In section 2 we introduce the basic concepts and define the cohomology of join dependencies systems of relational databases. In section 3 some theorems of homology theory are described. Section 4 applies these theorems and concepts for the theorem generalized of [3] about acyclic database schemes. At last, we reduced the problem dynamic integrity constraints of global and local databases to the well-known problem in topology. Section 5 describes topics for future research, in particular, describes generalizations of the cohomology of database schemes for other types of integrity constraints and for schemes of object-oriented databases.

The author is grateful to professor M.S.Tsalenko for constant attention to this work.

2 Some notions and definitions

Let us consider a relational model of a database.

Suppose $U = \{a_1, ..., a_n\}$ is a set of attributes. Let $\mathbf{C}$ be a category of subsets of U. Objects of $\mathbf{C}$ are subsets of U. If $X \subset Y \subset U$ and $X, Y \in Ob\mathbf{C}$, then $\alpha : X \to Y$ is a morphism of $\mathbf{C}$. Suppose for any $X, Y \in Ob\mathbf{C}$ there exist $X \cap Y \in Ob\mathbf{C}$ and $X \cup Y \in Ob\mathbf{C}$.

Let S be a database state over the category of attributes $\mathbf{C}$. The database state S is a functor from $\mathbf{C}^{op}$ to the finite sets category $FinSet$. Indeed, the functor S takes each set of attributes $X \in Ob\mathbf{C}$ to the set of rows of the table (or the relation) $S(X)$. If $\alpha : X \subset Y$ is a morphism of $\mathbf{C}$, then $S(\alpha) : S(Y) \to S(X)$ is a map of the tables. The map $S(\alpha)$ takes each row $t \in S(Y)$ to the row $S(\alpha)(t) = t|_X$, called the restriction of t to $X \subset Y$.

Note that in category theory functors from $\mathbf{C}^{op}$ are called presheaves over $\mathbf{C}$.

Suppose $X_1, ..., X_k$ are objects of $\mathbf{C}$, $X = \bigcup\limits_{i=1}^{k} X_i$ is their union, and $\alpha_{i,i';i} : X_i \cap X_{i'} \to X_i, \quad i, i' = 1, ..., k$ are morphisms of $\mathbf{C}$.

As usually [3], the joint operation $*$ takes the tables $S(X_1)$, ..., $S(X_k)$ to the table $S(X_1) * ... * S(X_k)$ with the set of attributes X.

The table $S(X_1) * ... * S(X_k)$ consists of all rows $(r_1, ..., r_k)$ such that $r_1 \in S(X_1), ..., r_k \in S(X_k)$ and $S(\alpha_{i,i';i})(r_i) = S(\alpha_{i,i';i'})(r_{i'})$ for $i, i' = 1, ..., k$.

We say that a join dependency $X_1, ..., X_n$ holds in a database state S if $S(X) = S(X_1) * ... * S(X_n)$.

Suppose integrity constraints of relational databases are a set of join dependencies.

Let $P = \{\{X_i^j : i \in I^j\} : j \in \{1, ...s\}\}$ be a set of join dependencies, where X_i^j is a object of the category $\mathbf{C}$. By $P(X)$ denote the set of all elements $\{X_i^j : i \in I^j\} \in P$ such that $X = \bigcup_{i \in I^j} X_i^j$. The set $P(X)$ is called the set of coverings of X.

Suppose P satisfies the following conditions:

- (i) for any set of attributes $X \in \mathbf{C}$, it follows that $\{X\} \in P(X)$;
- (ii) if $Y \to X$ is a morphism of the category $\mathbf{C}$ and $\{X_i : i \in I\}$ is a covering of X from $P(X)$, then $\{Y \cap X_i : i \in I\} \in P(Y)$;
- (iii) if $\{X_i : i \in I\} \in P(X)$ and $\{Y_{ij} : i \in J_i\} \in P(X_i)$ for any $i \in I$, then $\{Y_{ij} : j \in J_i, i \in I\} \in P(X)$.

A subset $\Sigma \subset P$ is called a set of generators of P if $P = P_\Sigma$ is the minimal set containing Σ and satisfies conditions $(i), (ii), (iii)$.

A category of attributes $\mathbf{C}$ together with a set of join dependencies Σ is called a scheme of a database and is denoted by Sch. A state $\mathcal{S}$ of the scheme Sch is a database state over the category of attributes $\mathbf{C}$ if the join dependencies P_Σ hold in $\mathcal{S}$.

Note that the conditions $(i), (ii), (iii)$ correspond in category theory to the definition of the Grothendieck pretopology P over the category $\mathbf{C}$ (see [8]). The set of states of the scheme Sch is the set of sheaves over the pretopology P_Σ.

Let $\mathcal{N}_1 = \{X_i : i \in I\}$ and $\mathcal{N}_2 = \{Y_j : i \in J\}$ be coverings from $P(X)$; then from conditions $(ii), (iii)$ it follows that $\{X_i \cap Y_j : i \in I, j \in J\} \in P(X)$. This covering is called the refinement of $\mathcal{N}_1$ и $\mathcal{N}_2$.

Therefore the set $P(X)$ is directed set under the relation of the refinement. Since $P(X)$ is a finite set, then there exist the minimal element in $P(X)$. This element of $P(X)$ is denoted by $\mathcal{M}_X$.

Theorem 1. *Let the covering* $\{X_i : i \in I\}$ *be an element of* $P(X)$*; then* $\mathcal{M}_X = \bigcup_{i \in I} \mathcal{M}_{Xi}$.

This theorem follows from conditions $(ii), (iii)$.

Let $\mathcal{S}$ and $\mathcal{S}'$ be two states of Sch without the condition of finiteness of $\mathcal{S}(X)$ and $\mathcal{S}'(X)$. A map $p : \mathcal{S} \to \mathcal{S}'$ is a natural transformation from the functor $\mathcal{S}$ to the functor $\mathcal{S}'$. By $Hom(\mathcal{S}, \mathcal{S}')$ denote the set of maps from $\mathcal{S}$ to $\mathcal{S}'$. Therefore the states of the scheme Sch and the states maps are the category of sheaves $\mathcal{S}h(Sch)$ over the Grothendieck pretopology P_Σ.

Hence all constructions applied for investigations Grothendieck topologies can be applied to the investigation databases schemes. For example, we shall use definitions of cohomology groups of the Grothendieck topology (see [8]).

Let K be a ring, let Mod_K be the category of K-modules. By $\mathcal{S}h_K(Sch)$ denote a subcategory of $\mathcal{S}h(Sch)$ consisted of sheaves over $\mathbf{C}$ to Mod_K. Thus an element $\mathcal{A} \in \mathcal{S}h_K(Sch)$ is a state of Sch such that $\mathcal{A}(X)$ is a K-module for each set of attributes of $\mathbf{C}$ and restrictions maps $\mathcal{A}(\alpha) : \mathcal{A}(Y) \to \mathcal{A}(X)$ are K-homomorphisms for morphisms $\alpha : X \to Y$ of $\mathbf{C}$.

Suppose $\mathcal{S}$ is a state of the scheme Sch; then by $F_K(\mathcal{S})$ denote the sheaf of free K-modules such that $F_K(\mathcal{S})(X)$ is the free K-module generated by the set $\mathcal{S}(X)$ and $F_K(\mathcal{S})(\alpha) : F_K(\mathcal{S})(Y) \to F_K(\mathcal{S})(X)$ is the homomorphism of the free modules defined by the restriction map $\mathcal{S}(\alpha) : \mathcal{S}(Y) \to \mathcal{S}(X)$.

Obviously, $F_K(\mathcal{S})$ is a sheaf if $\mathcal{S}$ is a sheaf. The sheaf $F_K(\mathcal{S})$ is called the free sheaf generated by $\mathcal{S}$. The sheaf $\mathcal{S}$ is a subsheaf of $F_K(\mathcal{S})$. If $\mathcal{A}$ is a sheaf of K-modules over Sch and $Hom_K(F_K(\mathcal{S}),\ \mathcal{A})$ is the K-module of morphisms of the sheaves, then $Hom_K(F_K(\mathcal{S}),\ \mathcal{A})$ is naturally isomorphic to $Hom(\mathcal{S},\ \mathcal{A})$. That is why we say that the sheaf $\mathcal{S}$ is represented by $F_K(\mathcal{S})$ in the category $\mathcal{S}h_K(Sch)$.

Let X be a set of attributes of $\mathbf{C}$. By h_X denote a presheaf over $\mathbf{C}$ such that $h_X(Y) = \mathbf{C}(Y, X)$, where $\mathbf{C}(Y, X)$ is the set of morphisms from Y to X of the category $\mathbf{C}$. The functor h_X is called a representable functor. It is easy to prove that h_X is a sheaf.

Let S be a presheaf over $\mathbf{C}$. The Yoneda's lemma [12] asserts that there exists a bijection of the set of sheaves maps $Hom(h_X, S)$ onto $S(X)$.

By $\mathcal{K}_X = F_K(h_X)$ denote the free sheaf K-modules generated by h_X. By definition, for any sheaf $\mathcal{A}(X) \in \mathcal{S}h_K(Sch)$ the set of sheaves morphisms $Hom_K(\mathcal{K}_X, \mathcal{A}) = Hom_K(F_K(h_X),\ \mathcal{A})$ is isomorphic to $Hom(h_X, \mathcal{A}) = \mathcal{A}(X)$.

Following [8] we shall give the definition of a cohomology of Sch.

Definition 1. *Suppose Sch is a scheme of a database, $\mathcal{A}$ is a sheaf of K-modules over Sch, and $\mathcal{S}$ is a state (sheaf) of Sch; then the q-derived functor of the functor $Hom(\mathcal{S},-)$ from the category $\mathcal{S}h_K(Sch)$ to the category K-modules is called the q-cohomology $H^q(\mathcal{S},\mathcal{A})$ of the state $\mathcal{S}$ with coefficients in $\mathcal{A}$. In other notation, $H^q(\mathcal{S},\mathcal{A}) = Ext^q(F_K(\mathcal{S}),\mathcal{A})$. If X is a set of attributes of $\mathbf{C}$, then by definition, put $H^q(X,\mathcal{A}) = H^q(h_X,\mathcal{A})$ and $H^q(Sch,\mathcal{A}) = H^q(h_U,\mathcal{A})$. In particular, $H^0(X,\mathcal{A}) = \mathcal{A}(X)$.*

3 Some theorems of homology theory

Let A, B be sheaves of K-modules.

The Yoneda's theorem [12] gives a representation of $Ext^q(\mathcal{B},\mathcal{A})$ as the K-module of extensions of the sheaf $\mathcal{B}$ by $\mathcal{A}$.

A extension of the sheaf $\mathcal{B}$ by $\mathcal{A}$ is a sheaf of K-modules $\mathcal{C}$ such that there exist a short exact sequence

$$0 \to \mathcal{A} \to \mathcal{C} \to \mathcal{B} \to 0.$$

Two extensions $\mathcal{C}$ and $\mathcal{C}'$ are equivalent if there exist a K-isomorphism $\varphi : \mathcal{C} \to \mathcal{C}'$ of the sheaves such that the following diagram is commutative:

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \mathcal{A} & \longrightarrow & \mathcal{C} & \longrightarrow & \mathcal{B} & \longrightarrow & 0 \\ & id(\mathcal{A}) & \downarrow & \varphi & \downarrow & id(\mathcal{B}) & \downarrow & & \\ 0 & \longrightarrow & \mathcal{A} & \longrightarrow & \mathcal{C}' & \longrightarrow & \mathcal{B} & \longrightarrow & 0 \end{array}$$

In the language of schemes states the extension $\mathcal{C}$ corresponds to the state of Sch such that $\mathcal{A}$ is a substate of $\mathcal{C}$ and $\mathcal{B}$ is a factor state of $\mathcal{C}$ by $\mathcal{A}$.

Theorem 2.*(Yoneda [12]). The cohomology $H^1(\mathcal{S},\mathcal{A})$ is isomorphic to the K-module of all extensions*

$$0 \to \mathcal{A} \to \mathcal{C} \to F_K(\mathcal{S}) \to 0$$

of the sheaf $F_K(\mathcal{S})$ by $\mathcal{A}$ to within the equivalence.

Now let us consider the Čech cohomology (see [8],too). We need it for calculations of the cohomology of Sch.

Let $\mathcal{N} = \{X_i : i \in I\} \in P(X)$ be a covering, let A be a presheaf of K-modules over $\mathbf{C}$. Denote by $C^*(\mathcal{N}; A)$ the cochain complex of the covering $\mathcal{N}$ with coefficients in A, where

$$C^n(\mathcal{N};A) = \prod_{|\bar{s}|=n+1} A(X_{\bar{s}}), \quad \bar{s} = (i_0, ..., i_n), \quad X_{\bar{s}} = \bigcap_{i\in\bar{s}} X_i$$

and the differentials $d^n : C^n(\mathcal{N};A) \to C^{n+1}(\mathcal{N};A)$ are defined in the standard fashion.

For example,

$$C^0(\mathcal{N};A) = \prod_{i\in I} A(X_i), \quad C^1(\mathcal{N};A) = \prod_{i1,i2\in I} A(X_{i1}\cap X_{i2}),$$

$$C^2(\mathcal{N};A) = \prod_{i1,i2,i3\in I} A(X_{i1}\cap X_{i2}\cap X_{i3})$$

and the differentials d^0, d^1 is defined by equations:

$$pr_{i0,i1}\circ d^0 = A(\alpha_{i0,i1;i1}) - A(\alpha_{i0,i1;i0}),$$

$$pr_{i0,i1,i2}\circ d^1 = A(\alpha_{i0,i1,i2;i1,2}) - A(\alpha_{i0,i1,i2;i0,i2}) + A(\alpha_{i0,i1,i2;i0,i1}),$$

where

$$pr_{\bar{s}} : C^n(\mathcal{N};A) = \prod_{|\bar{s}|=n+1} A(X_{\bar{s}}) \longrightarrow A(X_{\bar{s}})$$

is the projection , $\circ$ is the composition operation of homomorphisms, and

$$\alpha_{\bar{s};\bar{s}'} : \bigcap_{i\in\bar{s}} X_i \longrightarrow \bigcap_{i\in\bar{s}'} X_i$$

is the morphism of the category $\mathbf{C}$ for $\bar{s}' \subset \bar{s}$.

Definition 2. *Let $\mathcal{N} = \{X_i : i \in I\} \in P(X)$ be a covering of X, let A be a presheaf of K-modules over $\mathbf{C}$. The Čech cohomology $\check{H}^q(\mathcal{N}, A)$ of the covering $\mathcal{N}$ with coefficients in A is the cohomology of the cochain complex $C^*(\mathcal{N};A)$.*

By Definition 2, it follows that $\check{H}^0(\mathcal{N}, A) = A(X)$, if A is a sheaf.

If A is a presheaf, then by $\check{\mathcal{H}}^0(A)$ denote the presheaf such that $\check{\mathcal{H}}^0(A)$ takes each set of attributes X from $\mathbf{C}$ to $\check{H}^0(\mathcal{M}_X, A)$, where $\mathcal{M}_X$ is the minimal covering of X.

Theorem 3. *For any presheaf A over the scheme Sch it follows that $\check{\mathcal{H}}^0(A)$ is a sheaf (state) over Sch associated to A. If A is a sheaf, then $A = \check{\mathcal{H}}^0(A)$.*

The first part of Theorem 3 follows from Theorem 1 and from the definition of $\check{\mathcal{H}}^0(A)$. The second part of Theorem 3 follows from the definitions of a sheaf and of a 0-dimensional Čech cohomology of Sch.

Theorem 4. *Suppose $\mathcal{M}_X$ is a minimal covering in P_Σ of the scheme Sch, Y is any element of $\mathcal{M}_X$, and $\mathcal{A}$ is a sheaf of K-modules; then $H^q(Y, \mathcal{A}) = 0$ for $q > 0$.*

Let Y be the set of attributes Y of the theorem.

First let us prove that $Y \in \mathcal{M}_Y$, where is the minimal covering of Y. Indeed, assume the converse: $Y \notin \mathcal{M}_Y$. Then from the conditions (iii) for $\mathcal{M}_Y$ and $\mathcal{M}_X$ it follows that the refinement of $\mathcal{M}_X$ has not Y. This contradicts that $\mathcal{M}_X$ is the minimal covering of X. Thus we have $Y \in \mathcal{M}_Y$.

Let F be a presheaf. By the definition of a 0-dimensional Čech cohomology, it follows from $Y \in \mathcal{M}_Y$ that $\check{H}^0(\mathcal{M}_Y, F) = F(Y)$.

Let $0 \to \mathcal{A}_1 \to \mathcal{A}_2 \to \mathcal{A}_3 \to 0$ be a short exact sequence of sheaves of K-modules. By definition, from the short exact sequence of sheaves it follows that there exists the short exact sequence of presheaves $0 \to \mathcal{A}_1 \to \mathcal{A}_2 \to F \to 0$ and the sheaf $\mathcal{A}_3$ is associated to F. Using Theorem 3, we get $\mathcal{A}_3(Y) = \check{H}^0(\mathcal{M}_Y, F) = F(Y)$. On the other hand, by the definition of a short exact sequence of presheaves, we have the short exact sequence of K-modules

$$0 \to \mathcal{A}_1(Y) \to \mathcal{A}_2(Y) \to F(Y) \to 0.$$

If we combine this with the last equality, we get

$$0 \to \mathcal{A}_1(Y) \to \mathcal{A}_2(Y) \to \mathcal{A}_3(Y) \to 0.$$

We see that the functor

$$Hom(h_Y, \mathcal{A}) : \mathcal{A} \longmapsto \mathcal{A}(Y)$$

is exact on the category of sheaves over Sch. Since $H^q(Y, \mathcal{A})$ is the derived functor of the functor $Hom(h_Y, -)$ (by Definition 1), it follows that $H^q(Y, \mathcal{A}) = 0$ for $q > 0$. This completes the proof of Theorem 4.

Combining Theorem 4 with H. Cartan's theorem (see [8]) about acyclic coverings, we get the next theorem.

Theorem 5. *Suppose $X \in \mathbf{C}$ is a set of attributes and $\mathcal{M}_X$ is its minimal covering such that $X_i \cap X_j \in \mathcal{M}_X$ if $X_i, X_j \in \mathcal{M}_X$. Then for any sheaf of K-modules $\mathcal{A}$ over Sch, it follows that $H^*(X, \mathcal{A}) = \check{H}^*(\mathcal{M}_X, \mathcal{A})$.*

4 Some applications

In the paper [3], it is shown that for any state $\mathcal{S}$ of the database scheme Sch, it follows that the restrictions maps $\mathcal{S}(U) \to \mathcal{S}(X)$ are surjective iff Sch is acyclic. Let us consider this result using cohomology methods.

Let X_1 be a set of attributes and let A be a presheaf of K-modules. Denote by $A|_{X_1}$ the presheaf that takes each $Y \in Ob\mathbf{C}$ to $A|_{X_1}(Y) = A(X_1 \cap Y)$. Let $p : A \to A|_{X_1}$ be the morphism of the presheaves such that $p_Y : A(Y) \to A|_{X_1}(Y) = A(X_1 \cap Y)$ is the restriction homomorphism of A. Denote by $A_{\bar{X}_1}$ the presheaf that takes each Y to $A_{\bar{X}_1}(Y) = ker(p_Y)$, where $ker(p_Y) = \{a \in A(Y) | p_Y(a) = 0\}$. We have the following diagram of presheaves:

$$0 \to A_{\bar{X}_1} \to A \xrightarrow{p} A|_{X_1}.$$

Theorem 6. *Let $\mathcal{N} = \{X_1, ..., X_n\}$ be a covering of X, let A be a presheaf of K-modules such that $A(X) = A(X_1) * ... * A(X_n)$. Suppose $r_i \in A(X_i)$, $i = 1, ..., n$ such that the restrictions r_1 and r_i to $X_1 \cap X_i$, $i = 2, ..., n$ coincide. Then there exists a unique $h(r_1) \in \check{H}^1(\mathcal{N}; A_{\bar{X}_1})$ such that the element r_1 is extended to $r \in A(X)$ (r_1 is the restriction of r) iff $h(r_1) = 0$. The element $h(r_1)$ is called the obstruction of extensions of r_1 to $X \supset X_1$.*

This theorem generalizes results of [3]. It is easy to prove that if Sch is acyclic, then $\check{H}^1(\mathcal{N}; A_{\bar{X}_1}) = 0$ for any presheaf A.

Theorem 6 can be proved by direct calculations.

Indeed, let us consider $c = r_1 + r_2 + ... + r_n \in C^0(\mathcal{N}; A)$ and its coboundary $b = d_0(c)$. By definition, the value of b at $X_j \cap X_i$ equals $r_i - r_j$. By the condition of Theorem 6, the restrictions of $r_i - r_j$ to $X_1 \cap X_j \cap X_i$ equal 0. Hence the restriction $z_{j,i}$ of $r_i - r_j$ to $X_j \cap X_i$ is the element of $A_{\bar{X}_1}(X_j \cap X_i)$ and $z = \sum_{j,i=0}^{n} z_{j,i}$ is the element of $C^1(\mathcal{N}; A_{\bar{X}_1})$. Clearly, z is a cocycle and it defines $h(r_1) \in \check{H}^1(\mathcal{N}; A_{\bar{X}_1})$. If $h(r_1) = 0$, then $z = d_0(c')$ is a coboundary and $z_{j,i} = c'_i - c'_j$ over $X_j \cap X_i$, where $z_{j,i} = r_i - r_j$, and $c'_i \in A_{\bar{X}_1}(X_i)$. Now consider $r'_i = r_i - c_i$, $i = 1, ..., n$. We see that $r_1 = r'_1$ and the restrictions of r'_i and r'_j to $X_j \cap X_i$ coincide. Hence and from $A(X) = A(X_1) * ... * A(X_n)$ it follows that there exists $r \in A(X)$ such that the restriction of r to X_1 is r_1. This completes the proof of Theorem 6.

Another example of applications homology theory is connected with a concept of a map of database schemes and with a concept of dynamic integrity constrains.

Suppose $Sch = (U, \mathbf{C}, P)$ is a scheme of a global database, $Sch' = (U', \mathbf{C}', P')$ is a scheme of a local database (or view), and $\mathcal{S}h(Sch)$, $\mathcal{S}h(Sch')$ are their sheaves categories.

For example, if a global database consists from disconnected local databases, then its scheme is $Sch = (U, \mathbf{C}, P)$, $U = \bigcup_{i\in I} U_i$, $\mathbf{C} = \bigcup_{i\in I} \mathbf{C}_i$, $P = \bigcup_{i\in I} P_i$, where $Sch_i = (U_i, \mathbf{C}_i, P_i)$, $i \in I$ are schemes of local databases and $\bigcup_{i\in I}$ is the operation of disconnected union.

Definition 3. *A map of schemes $f : Sch' \to Sch$ is a function $f : U' \to U$ such that $f^{-1}(X) \in Ob\mathcal{C}'$ for any $X \in Ob\mathcal{C}$ and $\{f^{-1}(X_i) : i \in I\} \in P'(f^{-1}(X))$ for any covering $\{X_i : i \in I\} \in P(X)$. The functor $f^* : \mathcal{S}h(Sch) \to \mathcal{S}h(Sch')$ takes the state $\mathcal{S} \in \mathcal{S}h(Sch)$ of the global database to the state $\mathcal{S}' = f^*\mathcal{S} \in \mathcal{S}h(Sch)$ of the local database. The sheaf $f^*\mathcal{S}$ is the associated sheaf with the presheaf $X' \mapsto \mathcal{S}(f(X'))$, $X' \in Ob\mathcal{C}'$. The functor $f_* : \mathcal{S}h(Sch') \to \mathcal{S}h(Sch)$ takes the state $\mathcal{S}' \in \mathcal{S}h(Sch')$ to $f_*\mathcal{S}' \in \mathcal{S}h(Sch)$ such that $f_*\mathcal{S}'(X) = \mathcal{S}(f^{-1}(X))$.*

Note that the definition of the map f of schemes corresponds in category theory (see [8]) to the definition of the geometric morphism $f : \mathcal{S}h(Sch') \to \mathcal{S}h(Sch)$ of the sheaves categories.

The next theorem is the theorem of Grothendieck topology theory (see [8]).

Theorem 7. *Suppose $f : Sch' \to Sch$ is a map of schemes, $\mathcal{A}$ is a sheaf of K-modules over Sch; then there exist homomorphisms $f^q : H^q(Sch; \mathcal{A}) \to H^q(Sch'; f^*\mathcal{A})$ for any q. The homomorphisms f^q is functorial with respect to f and natural with respect to $\mathcal{A}$.*

Let $j^* : \mathcal{R} \subset \mathcal{S}h(Sch') \times \mathcal{S}h(Sch)$ be a functor is defined by a dynamic integrity constraints. If $(\mathcal{S}', \mathcal{S}) \in \mathcal{R}$, then $\mathcal{S}'$ is called an admissible new state of the local scheme Sch' at the state $\mathcal{S}$ of the global scheme Sch.

Suppose the functor j^* corresponds to the geometric morphism

$$j : \mathcal{S}h(Sch') \times \mathcal{S}h(Sch) \to \mathcal{R}.$$

Suppose also that there exists a unique geometric morphism $p : \mathcal{R} \to \mathcal{S}h(Sch)$ such that $p^*(\mathcal{S}) = (f^*\mathcal{S}, \mathcal{S})$ for all $\mathcal{S} \in \mathcal{S}h(Sch)$. Therefore the state $f^*\mathcal{S}$ is the admissible invariable state of Sch' at the state $\mathcal{S}$ of Sch.

Definition 4. *Let $f : Sch' \to Sch$ be a map of schemes. The dynamic integrity constraints $j : \mathcal{S}h(Sch') \times \mathcal{S}h(Sch) \to \mathcal{R}$ is called well defined if there exists a geometric morphism $d : \mathcal{S}h(Sch) \to \mathcal{R}$ such that $f^*(d^*(\mathcal{S}', \mathcal{S})) = \mathcal{S}'$ for all $(\mathcal{S}', \mathcal{S}) \in \mathcal{R}$. The functor $d^* : \mathcal{R} \to \mathcal{S}h(Sch)$ is called the change operation of states of the global scheme under admissible changes of states of the local scheme.*

Let $i_1 : Sch' \to Sch' \cup Sch$ be the inclusion. The functor $i_1^* : \mathcal{S}h(Sch') \times \mathcal{S}h(Sch) \to \mathcal{S}h(Sch')$ is the projection. By $d' : \mathcal{S}' \to \mathcal{R}$ denote the compositions of the geometric morphisms i_1 and j. Let us consider the following commutative diagram of geometric morphisms:

$$\begin{array}{ccc} \mathcal{S}h(Sch') & \xrightarrow{d'} & \mathcal{R} \\ || & & \downarrow p \\ \mathcal{S}h(Sch') & \xrightarrow{f} & \mathcal{S}h(Sch) \end{array}$$

Now we see that the problem: " Is $\mathcal{R}$ well-define dynamic integrity constraints?" is reduced to the well-known problem in topology theory about existence of the geometric morphism $d : \mathcal{S}h(Sch) \to \mathcal{R}$ such that $d' = d \circ f$ and $p \circ d = id(Sch)$, where $id(Sch)$ is the identical map of Sch. For the solution of this problem we can exploit the homology theory methods (see [6, 7]).

5 Generalization and topics for future research

We considered the case, when integrity constraints of relational databases were a set of join dependencies, but we can generalize these results for other more interesting cases.

In the general case, integrity constraints of relational databases are a set of sequents $\varphi \Rightarrow \psi$, where φ, ψ are formulas of a finite geometric language $\mathbf{L}$ (see the definition of the finite geometric language in [8]). Note that the language $\mathbf{L}$ is a many-sorted first-order predicate language with equality and without the symbols $\neg$, $\to$, and $\forall$. Now suppose that a scheme Sch is defined by a finite geometric theory $\mathbf{T} = (\mathbf{L}, \mathbf{I})$, where $\mathbf{I}$ is a finite set of sequents. Examples of such schemes are schemes with sets of functional, join, key, inclusion, Horn dependencies. In this case, there define also the sheaves category over Sch and our discussion may be extended to such schemes of databases.

Other generalization is for schemes of object-oriented databases. Following [9, 10, 11, 4] we consider a next object-oriented model of a database and its categorical representation.

A database scheme Sch is represented by a Scheme Definition Language (SDL). As usually, the database scheme Sch consists of a set of generators Σ (names of types, names classes, names of functions and relations, names of some elements) and a set of conditional equations E (integrity constraints).

Suppose SDL contains a few predefined basic types such as BOOL, NAT, STRING, etc. and also predefined type constructors for records, set_of, unions, etc. Types are templates of data structures.

As an example of database scheme in the complex-object model, consider a modified fragment of an university database scheme [9, 10].

Example 1. Schemes of object-oriented databases.

Definition: PERSON
PersonIdentityNo : PERSON $\to$ INT;
Name: PERSON $\to$ STRING;
Age: PERSON $\to$ INT;
Children: PERSON $\to$ Set_of(PERSON);
equations: Var p_1, p_2 : PERSON;
IF PersonIdentityNo(p_1) = PersonIdentityNo (p_2) THEN $p_1 = p_2$;
EndDef

Definition: STUDENT
Isa: STUDENT $\hookrightarrow$ PERSON;
GroupNamber: STUDENT $\to$ INT;
Advisor: STUDENT $\to$ PROF;
....
EndDef

Definition: PROF
Isa: PROF $\hookrightarrow$ PERSON;
....
EndDef

In this example, PERSON, STUDENT, PROF, INT, STRING, Set_of(PERSON) are names of classes. PersonIdentityNo, Name, Age, Children, GroupNumber, Advisor are names of functions.

The expression Isa:$T_1 \hookrightarrow T_2$ notes that T_1 is a subclass of T_2 and there exist the function name $inclusion(T_1, T_2) : T_1 \to T_2$. If T is a name of class, then Isa:$T \hookrightarrow T$ and $inclusion(T, T) = id(T)$ is a name of the identity function on T.

Set_of(PERSON) is the concrete domain of the parameterized abstract data type Set_of(T), in which the parameter T = PERSON.

Let $f_1 : T_1 \to T_2$, $f_2 : T_3 \to T_4$ be names of functions and Isa:$T_3 \hookrightarrow T_2$; then we can construct the new function name $f_2 \circ f_1 : T_1 \to T_2$. It is the composition of f_1 and f_2. For example, if s:STUDENT, then Name$\circ$ Advisor(s) is a well-defined term.

Such definition modification of the composition reflect the inheritance property of object-oriented database [1].

In category theory there are many definitions of constructions in which don't use elements of domains and which are analogs of set theory constructions. Examples of this constructions are the cartesian product, set of subsets, the exponential of two domains, the equalizer of two (functions names) and so on [2]. All of them are defined by a set of operations names and a set of conditional equations. The operations build new objects (classes names) or new morphisms (function names) from other objects and morphisms. The conditional equations represent the meaning of these operations. These categorical constructions may be considered as parameterized data types (as set_of(T))[5].

For any such database scheme $Sch = (\Sigma, E)$ there exists the initial (free) category $Cat(Sch)$ with the categorical operations, defined by the set of generators Σ and the set of conditional equations E. As usually, the category $Cat(Sch)$ is the quotient term algebra $T_\Sigma / \equiv_E$, where T_Σ is the set of all terms generated by names from Σ and $\equiv_E$ is the congruence generated by E on T_Σ.

In our case, the algebra $Cat(Sch)$ is the category with marking subobjects (ordered objects) and the modifying composition operation. On the other hand, if $Cat(Sch)$ has the categorical operations and equations of a topos, then $Cat(Sch)$ is a topos, and we can also use the homological methods for investigation the category $Cat(Sch)$.

6 Conclusion

In this paper, I tried to show that many problems of theory database may be translated to well-known problems of category theory and homology theory.

We defined the cohomology of schemes of relational databases with join dependencies. The cohomology is algebraic invariants of schemes. At last, we reduced the problem of dynamic integrity constraints to the well-known problem of topology.

As topics of future research we considered other types integrity constraints and schemes of object-oriented databases.

I hope that homology theory, which is useful for investigations of topological spaces, will be useful for investigations of databases schemes.

References

[1] Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S. *The Object-Oriented Database System Manifesto*, Proc. 1st DOOD, Kyoto, 1989

[2] Barr M., Wells C. *Category Theory for Computing Science*, Prentice Hall, 1990

[3] Beeri C., Fagin R., Maier D., Mendelzon A.O., Ullman J.D., Yannakakis M. *Properties of Acyclic Database Schemes*, Proc. of 13^{th} Ann. ACM Symp. on the Theory of Computing, 1981, pp.355–362

[4] Beniaminov E. M. *Foundation of Categorical Approach to Knowledge Representation Isv. Acad. Nauk USSR, Technical cybernet., 1988, N2, pp.21–33 (in Russian)*

[5] Ehrig H., Mahr B. Fundamentals of Algebraic Specification 1, Springer-Verlag, 1985

[6] Hilton P.J., Wylie S. Homology Theory, Cambridge, Massachusetts,1961

[7] Hu S.T. Homotopy Theory, New York, 1959

[8] Johnstone P.T. Topos theory, London: Academic Press INC., 1977

[9] Schewe K.-D., Thalheim B.Fundamental Concepts of Object Oriented Database Acta Cybernetica, v.11, N1-2, Szeged, 1993, pp.49–81

[10] Tuijn C., Gyssens M., Paredaens J. A Categorical Approach to Object-Oriented Data Modeling. In: Proc. of Third Workshop

on Foundations of Models and Languages for Data and Objects, Aigen, 1991, pp.187–196

[11] *Tuijn C., Gyssens M. Views and Decompositions of Databases from a Categorical Perspective, LNCS, 646, Springer-Verlag, 1992, pp.99–112*

[12] *Yoneda N. On the homology theory of modules, J. Fac. Sci. Tokyo, Sec. 1, 1954, 7, pp. 193–227*