

Rolik O.I.

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Ulianytska K.O.

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Amons O.A.

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Boiko O.V.

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Tsymbal S.I.

National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

MODERN INTERPRETATION FOR DATABASE DEVELOPMENT OF THE FIFTH NORMAL FORM

This article is devoted to the challenges of entity and relationship modeling within the framework of entity-attribute relationship analysis and further database normalization. The article reveals the theoretical and practical issues that arise when the use of secondary keys and attribute rules during infological design leads to violations of the Fifth Normal Form (5NF). It is shown that such violations can significantly affect the scalability of the database and complicate the denormalization process aimed at optimizing performance in practical applications. Particular attention is paid to the elimination of redundancy and insertion-update-deletion anomalies, which are frequently encountered in complex data models. The article presents a modern interpretation of the Fifth Normal Form by introducing the concept of conditional attributes. These conditional attributes allow developers to describe dependencies more precisely, without fragmenting data into an excessive number of related tables. The proposed approach helps avoid overcomplication of the schema and supports the development of flexible and maintainable database structures. It is emphasized that nontrivial dependencies often emerge not at the initial stage of development, but later—when business logic evolves and new requirements are introduced. It is clarified that insufficient time devoted to conceptual modeling is a major reason why database schemas fail to comply with 5NF. The proposed solution demonstrates how introducing conditional attributes, together with appropriate integrity constraints or triggers, can help ensure compliance without sacrificing performance. The article concludes that using modern interpretations of 5NF allows for building more robust, scalable, and semantically clear data models, capable of evolving alongside the application's functional growth.

Key words: normalization, fifth normal form, multivalued dependencies, secondary keys, conditional attributes, information systems and technologies, databases.

Introduction. The way to build a relational model is well-known. Basic principles of Entity Relationship (ER) modeling are as follows [1]:

- create and classify all entities related to the developing area;

- add and classify all necessary attributes to each entity;

- follow the four rules of attributes;

- add and classify all necessary relationships;

- generate ER model.

Each step of ER modeling is relatively simple, except for the first stage of each development – subject area analysis. The result of such analysis is creating an entity list; the main goal is to investigate each entity and select attributes. By following the rules of attributes, the possibility of redundancy existence should be decreased. So, redundancy in this case will be eliminated and only logical dependency might still be present.

Four rules of attributes have modern interpretation, which indicates more severe conditions for the attribute itself and dependency among them. In fact, some of these rules are very similar to the second and third normal forms.

In the next part of the article, attribute rules will be shown. Knowing the definition of the first five normal forms, including Boyce-Codd normal form, an interesting conclusion can be drawn. The ER model will follow the first five normal forms, including Boyce-Codd normal form in any case if we follow the four rules of attributes (including modern definition of these rules [2]). This conclusion leads us to a full investigation of the old interpretation of Fifth normal form (5NF).

Literature review and problem statement. In [1] a model based on N-ary relations, a normal form for database relations, and the concept of a universal data sublanguage are introduced and certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model. In [2] Chen's Entity-Relationship (ER) model with some enhancements needed for a better conceptual representation are used. This model is extensively used in many design methodologies, has an effective graphic representation, and is the de facto standard of most automatic tools for supporting database design. A joint methodology for conceptual database design and functional analysis are used. The activity of modeling involves the creation of relations, attributes, and indicating relationships among relations with a set of integrity constraints [5]. Modern interpretation of Four attribute rules, mentioned in [2] and [5] present a refined and updated approach to conceptual modeling, including [2-5]:

1. Clarity in attribute naming and meaning (semantic precision).
2. Atomicity (each attribute should represent one fact).
3. Avoidance of redundancy (removal of derived or duplicate attributes).
4. Stability and relevance (attributes should reflect intrinsic and stable properties of entities).

It builds on foundational principles but adapts them to modern practices in database design and information systems development. Using this interpretation reveals more and more facts that correct conceptual modeling leads us to follow most normal forms and some of them become useless in their old definition [6, 7].

The issues of ER modeling in the context of entities and attributes in the process of further normalization remain insufficiently researched. This requires eliminating the situation which leads to the violation of the Fifth normal form (5NF) when using secondary keys and rules of attributes during infological designing. This violation might be challenging during further database growing, when implementing denormalization steps to improve query execution in some cases takes place.

The aim and objectives of the study. The purpose of the investigation: is to develop a modern interpretation of 5NF for database development process.

To achieve the goal, it is necessary to perform the following tasks:

- to perform an analysis of attribute rules and secondary keys. Here will be found cause of 5NF violation and make assumption with new term – conditional attribute – for making investigation with new conditional attributes in relational model to check whether 5NF is followed;
- to propose a modern interpretation of the 5NF form to eliminate the situation which leads to a violation of the 5NF, when using secondary keys, conditional attributes and rules of attributes from infological designing.

The study materials and methods for database development of the fifth normal form.

Attribute rules and secondary keys

A relational database is a database where data has a structure, built on the relational model, in other words, it is a set of relations (tables) followed by severe rules (normal forms). Inside of each table, exists a set of attributes, which describes a specific entity, all attributes must be connected with each other in a one very specific way – they all depend on a primary key.

Figure 1 demonstrates several tables, which have primary key in a different manner. The most popular way – is using composite primary key as it shown on Figure 1, a). Such approach makes table follow by several normal forms – Second normal form, Boyce-Codd normal form and Fourth normal form (4NF). The other way, which is also frequently used – is using additional attributes as compound primary (Figure 1, b)). It might be used only in that case, when this table is not been referenced by another tables.

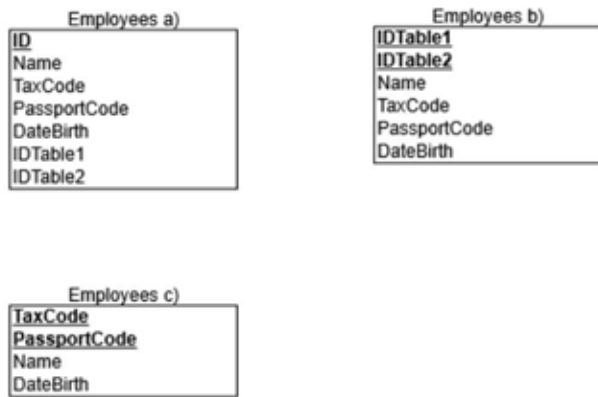


Fig. 1. Different approaches of creating primary key

Figure 1, c) demonstrates old approach, when developers use business attributes as part of compound primary key. Such approach leads us to a crucial redundancy; while creating relationships between such tables, we are duplicating business attributes (TaxCode and PassportCode) as they are used as foreign key in another tables.

A foreign key (or secondary key) is a combination of one or more additional attributes that is expressed in two or more relationships. It is used to create relationships between a pair of tables.

The normalization process is a formal technique that allows obtaining a relational data model, whose elements will meet the specific requirements of this model, resulting from the definition of normal forms.

Compliance with the requirements of normal forms is somehow necessary – it turns out that if the tables included in the database do not meet such requirements, i.e. the database schema is incorrect, then in certain situations there will be irregularities that prevent proper work with the database.

The normalization process is based on establishing functional dependencies (determination relationships) between attributes in a relation. We say that attribute A is functionally dependent on attribute B if 2 each value of B is related to exactly one value of A – in other words, A determines B . Furthermore, we say that attribute Z is transitively functionally dependent on attribute X if Z functionally depends on a subset of Y , Y is functionally dependent on X , while X is not functionally dependent on Y and Y is not functionally dependent on Z .

Example for such dependency is shown on Figure 2:

Functional Dependencies:

1) EmployeeID $\rightarrow$ EmployeeName, DepartmentID – each employee has exactly one name and is in one department;

	EmployeeID	EmployeeName	DepartmentName	DepartmentLocation
1	1	Alice	HR	New York
2	2	Bob	IT	San Francisco
3	3	Charlie	HR	New York

Fig. 2. Table Employees with functional dependency

2) DepartmentID $\rightarrow$ DepartmentName, DepartmentLocation – each department has one name and location.

For this case, to make table Employees follow normal forms, need to perform decomposition without any detailing (Figure 3):

```
CREATE TABLE Department (
    DepartmentID INT PRIMARY KEY,
    DepartmentName NVARCHAR(100) NOT NULL,
    DepartmentLocation NVARCHAR(100) NOT NULL
);

CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    EmployeeName NVARCHAR(100) NOT NULL,
    DepartmentID INT NOT NULL,
    FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
);
```

Fig. 3. Scripts for decomposed table Employees

Entity relation model is built during infological (conceptual) modeling, which demands us following some rules. Basic rules for conceptual modeling – are Four attribute rules. According to the original text, rules are:

1. Atomicity. Each attribute should represent a single, indivisible unit of data [3].

2. Full Functional Dependence on the Entity Identifier. Every attribute should be fully functionally dependent on the primary key of the relation [3].

3. Redundancy Avoidance. Attributes should be chosen in such a way that data redundancy is minimized [3]. Attributes should not repeat the same information or duplicate data that can be derived in another way.

4. No Derived Attributes. A relation should not contain attributes that can be derived from other attributes in the same relation [3].

All mentioned rules are related to some of the normal form, so it is easy to start the next step of development process – datalogic modeling or physical modeling. After ER model was created, we need to adapt it to relational database model. Normalization process should take place. If developer follow mentioned attribute rules, normalization process will be very fast.

First attribute rule is very similar to 1NF definition – a relation is in 1NF if and only if the domain of each attribute contains only atomic (indivisible)

values, and the value of each attribute contains only a single value from that domain. (The ordering of rows is immaterial, and each item in a row must be atomic, that is, indivisible) [1]. One of the algorithms that allows you to transform a non-normalized data set into a table that meets the requirements of first normal form is to remove repeating groups by inserting the appropriate data into empty columns, i.e., duplicating data in rows containing repeating groups. However, this often leads to redundancy, so if more than one value is defined for an attribute at a given time, or there is more than one attribute with the same name, a new entity must be defined that is described by this attribute. But here in the first attribute rule we have a more severe interpretation – instead of a single attribute like *FullName*, it is better to use separate attributes: *LastName*, *FirstName*, *MiddleName*. This rule tells us **that an attribute must not have a complex structure**.

Second attribute rule is almost the same as 2NF. A relation is in 2NF if it is in 1NF, and every non-prime attribute is fully functionally dependent on the whole of every candidate key of the relation (not just a part of any composite key) [1]. This rule tells us, that **an entity should have an identifier (surrogate attribute)**; in this case all other attributes will depend only on this primary key.

It should be noted that the quoted definition applies to tables that have composite primary keys; in this case, it may happen that some columns will depend on the entire key, while others will depend on its components. The transformation from 1NF to 2NF will consist of removing partial functional dependencies, i.e., of dividing the data structure so that each of the non-key attributes depends fully on the primary key. In practice, this means that if an entity has a unique identifier consisting of more than one attribute and/or relationship, and if some other attribute depends only on part of this composite identifier, then the attribute and the part of the identifier on which it depends form the basis of a new entity (table).

A relation that is in 2NF and whose non-key attributes depend on a key directly, not transitively, satisfies the requirements of 3NF. This means that if an attribute of an entity depends on another attribute that is not part of the unique identifier, then these attributes should form the basis for a new entity that has a one-to-many relationship with the original entity. The unique identifier for the new entity is the attribute on which the second attribute depends.

In the context of relational databases, 3NF is a key design principle and data modeling standard that provides effective organization and normalization of data in a database.

4NF is related to another type of dependency between relation attributes, namely multivalued dependencies. A multivalued dependency between attributes *A*, *B*, and *C* of relation *R* occurs when for each value *A* there is a set of corresponding values *B* and a set of values *C* (sets of values *B* and *C* are independent of each other). A table meets the requirements of the fourth normal form if it does not contain many multivalued dependencies, but only a single one. For example, if we have a separate table in which we have employee identifier and assigned private and work telephone numbers (two separate columns), then it should be divided into two separate tables.

5NF concerns so-called join dependencies, which are a generalization of multi-valued dependencies. It concerns relations in which there are more than two multi-valued dependencies. In reality, situations requiring the application of the normalization procedure ensuring fifth normal form almost never occur, nevertheless, this form is presented in the literature on the subject. The general definition says that a relation is in fifth normal form if it is in fourth normal form and does not contain join dependencies, in other words, if there is no lossless decomposition into smaller tables.

The third and fourth rules are very similar to each other. But, starting from this point, we are approaching 5NF, toward understanding it. The third rule only points out that there is no need to duplicate information, for example (Figure 4), there is no need to keep the name of document type in table Documents, if we already have foreign key to table DocumentTypes.



Fig. 4. Database diagram with/without redundant attribute

To get the name of document type, we just need to JOIN concrete table (Figure 5).

This rule leads us to a very important constraint – secondary key (foreign key [4]). The whole relational database schema is based on the usage of pairs – primary and foreign keys. This is the main feature of all relational databases. An example of storing data using the described keys is presented in Figure 6.

```

1 SELECT doc.DocumentNumber, doc.DocumentDate, type.DocumentTypeName
2 FROM T_Documents doc
3 JOIN T_DocumentTypes type ON doc.IDDocumentType = type.ID
4

```

DocumentNumber	DocumentDate	DocumentTypeName
234/o	2025-05-27 16:20:28.907	Type1
101/u	2025-05-27 16:20:28.907	Type1
201/h	2025-05-27 16:20:28.907	Type2
301/o	2025-05-27 16:20:28.907	Type2
401/f	2025-05-27 16:20:28.907	Type1

Fig. 5. Results of query with JOIN expression

Primary key (DocumentType)		DocumentTypeName
1	1	Type1
2	2	Type2
3	3	Type3
4	4	Type4
5	5	Type5

ID	Foreign key (DocumentType)	DocumentNumber	DocumentDate
1	1	234/o	2025-05-27 16:20:28.907
2	1	101/u	2025-05-27 16:20:28.907
3	2	201/h	2025-05-27 16:20:28.907
4	2	301/o	2025-05-27 16:20:28.907
5	3	401/f	2025-05-27 16:20:28.907

Fig. 6. Example of storing the referenced data

The advantage of using secondary keys is that we can reference data as many times as needed (Figure 6), only the primary key will be used, not any business attributes (e.g., Name, DateOfBirth, Code, etc.). At any time, each business attribute can be changed, and we can access fresh data at any time through the reference. There is no redundancy, no duplicity, no multi-valued dependency and no non-trivial dependency (as it will appear in 5NF).

Modern interpretation of the Fifth normal form

The original definition of the 5NF, also known as Projection-Join Normal Form (PJNF), comes from Ronald Fagin in his seminal paper [8]: a relation R is in 5NF, or PJNF, if every join dependency in R is implied by the candidate keys of R .

C.J. Date in his work [3] gave for 5NF more details and descriptions: a relation variable R is in 5NF, also sometimes referred to as Projection-Join Normal Form, if and only if every non-trivial join dependency in the relation variable R is implied by the candidate key(s) of R , provided the following conditions are met:

a) a join dependency $\{A, B, \dots, Z\}$ in relation variable R is considered trivial if and only if at least one of the subsets $A, B, \dots, Z$ of the set of attributes equals the entire set of attributes of R .

b) a join dependency $\{A, B, \dots, Z\}$ in relation variable R is said to be implied by the candidate key(s) of R if and only if each of the subsets $A, B, \dots, Z$ of the set of attributes is a superkey for R .

The next Figure 7 illustrates tables (entities), that are not follow 5NF.

```

1 CREATE TABLE EmployeeSkillsLanguages (
2     Employee VARCHAR(50),
3     Skill VARCHAR(50),
4     Language VARCHAR(50),
5     PRIMARY KEY (Employee, Skill, Language)
6 );
7
8 INSERT INTO EmployeeSkillsLanguages (Employee, Skill, Language) VALUES
9 ('Alice', 'Java', 'English'),
10 ('Alice', 'Java', 'French'),
11 ('Alice', 'Python', 'English'),
12 ('Alice', 'Python', 'French');
13
14 select * from EmployeeSkillsLanguages

```

Employee	Skill	Language
Alice	Java	English
Alice	Java	French
Alice	Python	English
Alice	Python	French

Fig. 7. Relational table, that contains join dependencies

The example was taken from [3], but reframed using modern relational database design. This is the most common example of a table that does not follow 5NF. But, if we investigate such tables closely, we will see a conceptual model mistake. At the step of infological modeling, we need to get all appropriate entities and their relationships. Obviously, the table EmployeeSkillsLanguage is not an entity from the ER model. Such tables represent an associative entity with (in this case) three foreign keys, and if we have dependencies between attributes Employee, Skill, Language – it is described at the conceptual step, add two more entities or any other integrity constraints further on physical modeling (e.g., CHECK [9]).

Generally, to make the EmployeeSkillsLanguages table follow 5NF, we need to apply decomposition as shown in Figure 8.

This is bad practice in modern development. Today, developers are creating three separate vocab-

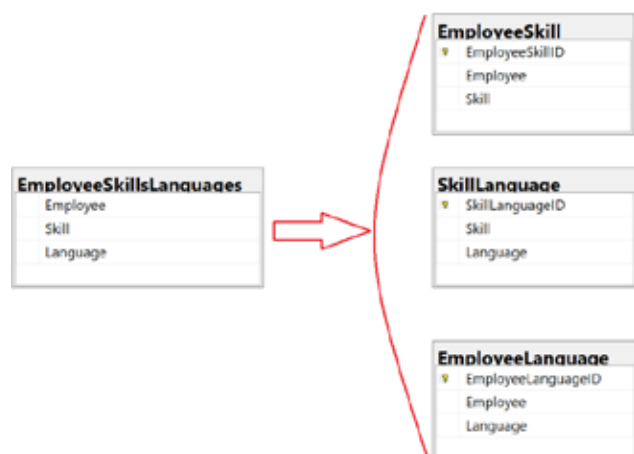


Fig. 8. Decomposition of table, which is not following 5NF

ulary tables (Employees, Skills, Languages) and any business rule can be put in the associative table as an integrity constraint CHECK or even in TRIGGER [9] programming code. Another best approach is during application development and writing documentation use cases that specify how to fill the associative table (application form) in the correct way.

An example of a relational model with foreign keys and dictionary tables is shown in Figure 6.

Theory tells us that the table Project (Figure 9) does not follow 5NF. But if we create tables like EmployeeSkill, SkillLanguage, EmployeeLanguage (Figure 8), even with modern foreign keys, it will increase complexity for application development.

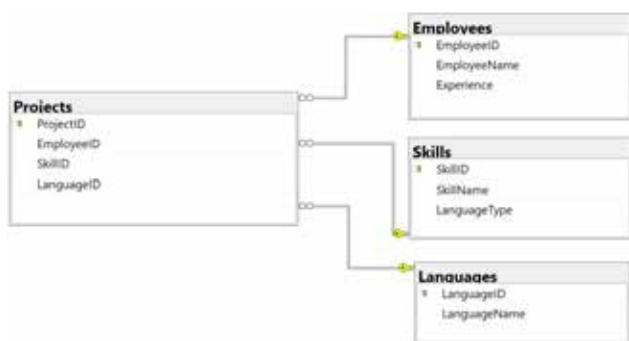


Fig. 9. Modern solution with secondary keys

To prevent development complexity and to make the table Projects follow 5NF, we may add a specific CHECK constraint or TRIGGER.

Assume that the dependency Employee $\twoheadrightarrow$ Skills can be handled by adding an attribute Experience, which indicates that an employee with a particular value of Experience={1,2}, should have appropriate Skills={S1,S2,...,Si}.

The same goes for Skills $\twoheadrightarrow$ Languages, this dependency can be handled by adding attribute LanguageType, which indicates, that skill with particular LanguageType={1,2}, should have appropriate Language={S1,S2,...,Sk}.

On Figure 10 and Figure 11 is shown example of trigger for Employee-Skills, Skills-Language dependencies.

This is one way of how to handle 5NF. There also might be applied check constraints or added algorithm to application code, to prevent dependency violation.

The main point is that situation on fig.6 would never appear if on conceptual modeling step specific attributes was added (ex, Employees.Experience; Skills.LanguageType etc.) for resolving dependency between entity, cause each dependency, in fact, based on some conditions, which might appear further in, as example, TRIGGER (Figure 10, Figure 11).

```
CREATE TRIGGER trg_Projects_Enforce5NF
ON Projects
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    -- Inserted projects
    DECLARE @EmployeeID INT, @SkillID INT, @LanguageID INT, @Experience TINYINT, @LanguageType TINYINT;

    -- Cursor for each row is inserted (if multi-row insert)
    DECLARE cur CURSOR FOR
    SELECT i.EmployeeID, i.SkillID, i.LanguageID, e.Experience, s.LanguageType
    FROM Inserted i
    JOIN Employees e ON i.EmployeeID = e.EmployeeID
    JOIN Skills s ON i.SkillID = s.SkillID;

    OPEN cur
    FETCH NEXT FROM cur INTO @EmployeeID, @SkillID, @LanguageID, @Experience, @LanguageType;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- 1. validate Skills match Experience level
        IF @Experience = 1 AND @SkillID NOT IN (1, 2)
        BEGIN
            RAISERROR('Employee with Experience=1 must be assigned only Skills S1 or S2.', 16, 1)
            ROLLBACK TRANSACTION
            RETURN
        END
        ELSE IF @Experience = 2 AND @SkillID NOT IN (3, 4, 5)
        BEGIN
            RAISERROR('Employee with Experience=2 must be assigned only Skills S3, S4, or S5.', 16, 1)
            ROLLBACK TRANSACTION
            RETURN
        END
        -- 2. Validate Skill belongs to Employee
        IF NOT EXISTS (
            SELECT 1 FROM EmployeeSkills es
            WHERE es.EmployeeID = @EmployeeID AND es.SkillID = @SkillID
        )
        BEGIN
            RAISERROR('Employee does not possess the assigned Skill.', 16, 1)
            ROLLBACK TRANSACTION
            RETURN
        END
        -- 3. Validate language by Skill.LanguageType
        IF @LanguageType = 1 AND @LanguageID NOT IN (1, 2) -- L1, L2
        BEGIN
            RAISERROR('languageID must be L1 or L2 for LanguageType=1.', 16, 1)
            ROLLBACK TRANSACTION
            RETURN
        END
        ELSE IF @LanguageType = 2 AND @LanguageID NOT IN (3, 4, 5) -- L3, L4, L5
        BEGIN
            RAISERROR('languageID must be L3, L4, or L5 for LanguageType=2.', 16, 1)
            ROLLBACK TRANSACTION
            RETURN
        END
        FETCH NEXT FROM cur INTO @EmployeeID, @SkillID, @LanguageID, @Experience, @LanguageType;
    END

    CLOSE cur
    DEALLOCATE cur;

    -- All checks passed: perform insert
    INSERT INTO Projects (ProjectID, EmployeeID, SkillID, LanguageID)
    SELECT ProjectID, EmployeeID, SkillID, LanguageID FROM Inserted
END
```

Fig. 10. Trigger for Projects table for following 5NF (start)

```

-- 2. Validate Skill belongs to Employee
IF NOT EXISTS (
    SELECT 1 FROM EmployeeSkills es
    WHERE es.EmployeeID = @EmployeeID AND es.SkillID = @SkillID
)
BEGIN
    RAISERROR('Employee does not possess the assigned Skill.', 16, 1)
    ROLLBACK TRANSACTION
    RETURN
END

-- 3. Validate language by Skill.LanguageType
IF @LanguageType = 1 AND @LanguageID NOT IN (1, 2) -- L1, L2
BEGIN
    RAISERROR('languageID must be L1 or L2 for LanguageType=1.', 16, 1)
    ROLLBACK TRANSACTION
    RETURN
END
ELSE IF @LanguageType = 2 AND @LanguageID NOT IN (3, 4, 5) -- L3, L4, L5
BEGIN
    RAISERROR('languageID must be L3, L4, or L5 for LanguageType=2.', 16, 1)
    ROLLBACK TRANSACTION
    RETURN
END

FETCH NEXT FROM cur INTO @EmployeeID, @SkillID, @LanguageID, @Experience, @LanguageType;
END

CLOSE cur
DEALLOCATE cur;

-- All checks passed: perform insert
INSERT INTO Projects (ProjectID, EmployeeID, SkillID, LanguageID)
SELECT ProjectID, EmployeeID, SkillID, LanguageID FROM Inserted
END
```

Fig. 11. Trigger for Projects table for following 5NF (finish)

For preventing tables from nontrivial dependency, need to add in 5NF interpretation specific conditional attributes, that will be used in code, SQL script or SQL programming code to resolve such dependency.

Working in such way – any nontrivial dependency might be resolved even on database level.

RESULTS OF INVESTIGATING MODERN DEFINITION FOR 5NF

Based on the example and analysis which were applied in this article, the definition for Fifth Normal Form can be improved by including a new determination – conditional attributes.

The main problem of each table that might not be compliant with 5NF is the lack of time that developers spend on conceptual modeling.

The situation that appears in Figure 6 is not knowing the conditions which lead to dependency, in this case, nontrivial dependency, because all three attributes are involved in it.

Using conditional attributes that will significantly clarify dependency between sets of values, we can achieve compliance with 5NF. That means: getting rid of all kinds of anomalies (during running insert, update, delete operations in the database); getting rid of data redundancy (including using secondary keys) and implementing nontrivial dependency without harming future applications.

Old definition of 5NF: a relation R is in 5NF, or PJNF, if every join dependency in R is implied by the candidate keys of R .

Modern definition of 5NF, which includes conditional attributes: a relation R is in 5NF, or PJNF, if every join dependency in R is implied by the candidate keys of R and for each pair of dependent attributes a conditional attribute was added.

Conditional attribute – an attribute that leads us to understanding dependency between sets of values in each pair of dependent attributes.

DISCUSSION OF RESULTS OF INVESTIGATING MODERN DEFINITION FOR 5NF

This article is devoted to the issues of Entity Relationship modeling in the context of the ratio of entities

and attributes in the process of further normalization. The article shows the elimination of that situation which leads to the violation of the fifth normal form when using secondary keys and rules of attributes of infological design. This violation might be challenging in the further growth of the database itself with implementing attempts at denormalization steps to improve query execution in some cases.

After analysis and investigation that was made and also in the article, a modern interpretation of the Fifth Normal Form was proposed. This definition allows developers to better understand the normalization process and take into account some steps in conceptual modeling.

Conclusions. The material of this article is devoted to the normalization process, which apparently will take place not at the start of developing an application.

Such nontrivial dependency in tables frequently appears during improvement and adding new features, changing business rules of application usage.

Modern interpretation of 5NF allows developers to simplify resolving situations with nontrivial dependencies by adding new conditional attributes for each dependency, which may lead to not creating a bunch of tables, but adding new integrity constraints or program units such as triggers.

Bibliography:

1. Codd, E. F. A Relational Model of Data for Large Shared Data Banks / E. F. Codd // Communications of the ACM. – 1970. – Vol. 13, No. 6. – P. 377–387.
2. Batini, C., Ceri, S., Navathe, S. B. Conceptual Database Design: An Entity-Relationship Approach. – San Mateo, CA: Benjamin/Cummings Publishing, 2011. – ISBN 9780201325775.
3. Date, C. J. An Introduction to Database Systems. – 8th ed. – Boston : Pearson/Addison Wesley, 2003. – 1024 p.
4. Elmasri, R., Navathe, S. B. Fundamentals of Database Systems. – 7th ed. – Boston : Pearson, 2017. – ISBN 978-0133970777.
5. Elmasri, R., Navathe, S. B. Improving Conceptual Modeling with Semantic Clarity and Data Quality Constraints // Journal of Database Management (JDM). – 2021. – Vol. 32, No. 1. – P. 1–21. DOI: 10.4018/JDM.2021010101.
6. Rolik, O., Amons, O., Ulianytska, K., Kolesnik, V. Modernization of the Second Normal Form and Boyce-Codd Normal Form for Relational Theory // In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education III. ICCSEEA 2020. Advances in Intelligent Systems and Computing, vol. 1247. – Springer, 2021. – P. 296–305. DOI: 10.1007/978-3-030-55506-1_27.
7. Rolik, O., Ulianytska, K., Khmeliuk, M., Khmeliuk, V., Kolomiets, U. Increase efficiency of relational databases using instruments of second normal form // 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory, ATIT 2021 – Proceedings. – 2021. – P. 221–225.
8. Fagin, R. Normalization and Relational Database Operators // ACM SIGMOD Record. – 1979. – Vol. 6, No. 3. – P. 33–41. DOI: 10.1145/582095.582099.
9. Jorgensen, K. Microsoft SQL Server 2022: A Beginner's Guide. – Sixth Edition. – McGraw-Hill Education, 2022. – ISBN 9781264713886.

Ролік О.І., Ульяницька К.О., Амонс О.А., Бойко О.В., Цимбал С.І. СУЧАСНА ІНТЕРПРЕТАЦІЯ РОЗРОБКИ БАЗИ ДАНИХ П'ЯТОЇ НОРМАЛЬНОЇ ФОРМИ

Стаття присвячена проблемам моделювання сутностей і зв'язків у контексті аналізу типів сутність-атрибут і подальшої нормалізації баз даних. У статті розкрито теоретичні та прикладні аспекти, які виникають при використанні вторинних ключів і правил атрибутів в інфологічному проєктуванні, що може призводити до порушення п'ятої нормальної форми (5НФ). З'ясовано, що такі порушення негативно впливають на масштабованість бази даних та ускладнюють процес денормалізації, який здійснюється з метою оптимізації продуктивності при практичному використанні. Розкрито сучасний підхід до інтерпретації п'ятої нормальної форми, що ґрунтується на впровадженні поняття умовного атрибута. Запропоновано використовувати умовні атрибути для точнішого опису залежностей без надмірного розбиття даних на велику кількість взаємопов'язаних таблиць. Це дозволяє уникнути ускладнення схеми та забезпечити гнучкість і зручність підтримки бази даних у процесі її розвитку. Підкреслено, що складні залежності між атрибутами часто з'являються не на початковому етапі розробки, а під час модифікації бізнес-логіки та впровадження нових функціональних вимог. З'ясовано, що однією з головних причин невідповідності схем баз даних вимогам 5НФ є недостатній час, приділений концептуальному моделюванню. Запропоновано рішення, яке полягає у впровадженні умовних атрибутів разом із відповідними обмеженнями цілісності або тригерами, що дозволяє забезпечити відповідність 5НФ без шкоди для продуктивності. Стаття доводить, що сучасне трактування 5НФ дає змогу проєктувати більш надійні, масштабовані та семантично зрозумілі моделі даних, які можуть еволюціонувати відповідно до зростаючих вимог прикладного програмного забезпечення.

Ключові слова: нормалізація, п'ята нормальна форма, багатозначні залежності, вторинні ключі, умовні атрибути, інформаційні системи та технології, бази даних.

Дата надходження статті: 31.07.2025

Дата прийняття статті: 01.09.2025

Опубліковано: 27.10.2025