

Дормидошина Дарья Андреевна,
Технический директор, АО «ЦКБ «Дейтон»
Dormidoshina Daria A., Technical Director,
JSC «CDO «Deyton»

Рубцов Юрий Васильевич,
Генеральный директор, АО «ЦКБ «Дейтон»
Rubtsov Yuriy, General Director
JSC «CDO «Deyton»

**СОВЕРШЕНСТВОВАНИЕ ПРОЦЕССОВ
ВЫБОРА ЭЛЕКТРОННОЙ КОМПОНЕНТНОЙ БАЗЫ
ЧЕРЕЗ ВАЛИДАЦИЮ РЕЗУЛЬТАТОВ НОРМАЛИЗАЦИИ ДАННЫХ
В ПРОЦЕССАХ РАЗРАБОТКИ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ
IMPROVEMENT OF THE PROCESSES OF SELECTING
AN ELECTRONIC COMPONENT BASE THROUGH VALIDATION
OF THE RESULTS OF DATA NORMALIZATION IN THE PROCESSES
OF DEVELOPING RADIOELECTRONIC EQUIPMENT**

Аннотация. В статье рассматривается проблема выбора электронной компонентной базы при разработке радиоэлектронной аппаратуры, вызванная избыточностью и несогласованностью данных. Предлагается использование методов нормализации данных для повышения эффективности работы информационной системы. Описывается процедура валидации результатов нормализации данных, направленная на устранение слабых мест в структуре базы данных. Исследуются различные типы сбора информации и анализируются функциональные зависимости с применением ER-моделирования. В результате применения процедуры валидации результатов нормализации данных достигается улучшение качества данных, сокращение избыточности и повышение целостности информации. Полученные результаты подтверждают практическую ценность нормализации для оптимизации процессов выбора ЭКБ.

Abstract. The article considers the problem of choosing an electronic component base when developing radio-electronic equipment, caused by redundancy and inconsistency of data. It is proposed to use data normalization methods to improve the efficiency of the information system. The procedure for validating the results of data normalization aimed at eliminating weaknesses in the database structure is described. Various types of information collection are studied and functional dependencies are analyzed using ER modeling. As a result of using the procedure for validating the results of data normalization, the quality of data is improved, redundancy is reduced, and the integrity of information is increased. The results confirm the practical value of normalization for optimizing the processes of choosing an electronic component base.

Ключевые слова: Электронная компонентная база, радиоэлектронная аппаратура, база данных, валидация данных

Keywords: Electronic component base, radio-electronic equipment, database, data validation

Введение

В процессах разработки радиоэлектронной аппаратуры (РЭА) возникает острая потребность в оптимальном выборе изделий электронной компонентной базы (ЭКБ). Для этого нужна достоверная и актуальная информация о функциональном назначении изделий, области их применения, параметрах, показателях, конструктивных элементах и прочих данных.

Оптимальный выбор ЭКБ требует оптимальный сбор и обработку данных, применение методов их нормализации и верификацию результатов нормализации. В информации об ЭКБ имеется много избыточных данных. Основное последствие этой проблемы является то, что



невозможно или достаточно трудно найти необходимые данные. Из-за несогласованности форматов процесс выбора ЭКБ для применения приводит к потере нужной информации. В результате, выбор ЭКБ отнимает много времени, вызывает недовольство конструкторов РЭА. Данное исследование направлено на повышение эффективности выбора ЭКБ для разработки РЭА.

Нормализация – это процесс организации данных таким образом, чтобы свести к минимуму избыточность. Модель отношений лежит в основе концепции нормализации и нормальных форм. Нормализация – это процедурный способ организации базы данных (БД), которая играет важную роль в работе Системы. Нормализация предвосхищает проблемы, которые могут возникнуть – это избыточные данные и их несогласованность. Нормализация данных снижает опасность возникновения аномалий с данными.

БД реляционного типа состоит из нескольких логически связанных таблиц, каждая из которых визуально представляет электронную таблицу. Каждый столбец содержит одну и ту же по назначению информацию, например, такую как условные обозначения изделий, наименования изготовителей и пр. Каждая таблица будет иметь определенные атрибуты, указывающие на то, как она связана с другими таблицами. Связи между таблицами являются важным элементом в БД, так как они определяют отношения между данными.

Уровень нормализации элементов БД может быть выполнен за полиномиальное время с использованием активного набора функциональных зависимостей (NF) в терминах классов эквивалентности атрибутов, чтобы реляционная схема находилась на разных уровнях записи, таких как 1NF, 2NF, 3NF и нормальная форма Бойса-Кодда, 4NF, пятая нормальная форма (5NF), или нормальная форма проекции-соединения (PJ/NF) [1].

Тщательно продумывая и разрабатывая формы и таблицы БД можно свести к минимуму дублирование, уменьшить объёмы, упростить вставку информации, защитить данные для обеспечения их согласованности, выполнять более быстрые транзакции за счет обработки меньшего количества данных и формировать оптимальный набор данных об ЭКБ для их выбора.

Процесс нормализации обычно упрощает отношения и снижает опасность возникновения аномалий. Таким образом, метод нормализации повышает надежность хранения данных.

NF в теории реляционных БД представляют собой правила проектирования записей. Такие правила могут соответствовать для первого-пятого NF [2]. Процесс нормализации основан на анализе функциональных зависимостей. Концепция функциональной зависимости полезна при проектировании и анализе реляционной БД. Одним из ключевых аспектов функциональных зависимостей от реляционных БД является представление отношения в различных NF. Это играет важную роль в предотвращении таких проблем, как вставка, обновление или удаление данных, в зависимости от того, является ли операция, вызывающая ошибку, вставкой, обновлением или удалением строки соответственно. В большинстве практических работ последнего времени по NF БД учитывается только первичный ключ и игнорируются остальные потенциальные ключи. Это важно, потому что в нем подчеркивается детальный тщательный отбор и формальная модель в сочетании с техникой формального анализа.

Проблемы для оптимального выбора изделий

Качество БД с информацией об изделиях ЭКБ сталкивается с проблемами при импорте данных из внешних источников, это субъекты и объекты, идентифицирующие ее происхождение, доступные и дающие разрешение для ее использования системами и обладающие достаточной определённой. Информация достоверна, если она отражает истинное состояние изделия. Достоверную информацию получают от разработчиков и потребителей изделий ЭКБ, исследовательских, измерительных и испытательных организаций. Достоверная информация находится в конструкторской и технологической документации на изделия, результатах исследований, измерений, испытаний, применения, эксплуатации и модернизации, моделях [3].



К данным, которые не нормализованы, может относиться информация в различных масштабах единиц измерения, об одних и тех же параметрах, показателях, аналогах, других характеристиках [4]. Отсутствие нормализации в таких данных может быть опасно для работы Системы с точки зрения безопасности данных, использования памяти, эффективности обновления информации в БД и, возможно, самое главное, целостности данных. Информация до нормализации – это данные, которые не были логически разбиты на небольшие и управляемые таблицы. Целью данного исследования является разработка методов валидации результатов нормализации данных (ВРНД), а также распознавание и замена значимых полей в таблицах с целью повышения эффективности функционирования БД.

Описание программно-аппаратной среды исследований

Системы формирования предложений на выбор ЭКБ в процессах разработки РЭА (Система) создана с применением SQL Server 2019 и использования SQL (Structured Query Language – язык структурированных запросов). Это язык программирования, применяемый для создания, модификации и управления данными в реляционной БД, управляемой соответствующей системой управления БД.

С помощью PHP (Hypertext Preprocessor – препроцессор гипертекста) языка программирования, разработаны веб приложения Системы, которые встраиваются в HTML (HyperText Markup Language – стандартизированный язык гипертекстовой разметки документов для работы с веб-страницами).

Для клиентских и серверных приложений используется язык программирования JavaScript. Он расширяет возможности системы за счет применения большого количества библиотек и фреймворков. При этом, он имеет несогласованность браузерной поддержки: различные браузеры по-разному интерпретируют JavaScript-код. Имеются проблемы с безопасностью на клиентской стороне: JavaScript-код, выполняемый в браузере, может быть уязвим к атакам.

При этом, синхронная природа PHP вызывает задержки при обработке множества запросов. PHP используется на сервере, что требует дополнительных технологий оптимизации данных и PHP может быть уязвим к различным атакам. В Системе используется комбинация различных технологий программирования. Используется оптимальное решение – применение PHP и JavaScript, где каждая технология программирования должна применяться в том месте, где она приносит максимальный эффект [5].

Основная причина проблемы с Системой заключается в том, что специалисты не могут быстро и легко найти актуальные данные. Ошибки данных в процессе выбора ЭКБ создают неполные наборы и из-за этого конструкторы не получают от Системы ожидаемый эффект.

Анализ типов сбора информации

В этом исследовании используется подход в получении информации об ЭКБ от различных источников, чтобы найти и обнаружить слабые места в БД. Предполагается, что нормализация данных об ЭКБ для выявления значимых полей и предотвращения ошибок функционирования улучшит работу Системы.

При заполнении информации об ЭКБ разработчики изделий сталкиваются с проблемами. При этом, при поступлении информации об ЭКБ от разработчиков изделий, возникают проблемы в БД. Принято решение об использовании шаблонов – это инструменты для структурированного сбора данных [6], позволяющий регулярно собирать данные, запрашивать обновления, согласовывать и использовать данные для оптимального выбора ЭКБ.

Разработана открытая анкета, предназначенная для сбора позиций о структуре шаблона. В результате анализа полученных анкет, сформированы шаблоны и определены типы сбора информации об ЭКБ:

1. Сплошной сбор – это сбор всей информации об ЭКБ. При сплошном сборе ограничений по объёму информации не существует, но есть ограничения по времени. Это связано с тем, что данные в БД устаревают и затягивание времени со сбором, ведет к потере их актуальности.



Сплошной сбор позволяет, использовать методы сопоставления и иметь в БД актуальную информацию вплоть до ежедневной основы. Алгоритм методов может постоянно обновлять связи и принимать решения об идентичности ЭКБ на уровне Системы, не используя мануальный ручной труд для прописывания ссылок по каждому сопоставлению.

Сплошной сбор подразделяется на 2 подвида:

- сплошной сбор без целевых указаний о конкретных изделиях ЭКБ. В этом случае в шаблон заполняется информация о всей ЭКБ изготовителя;
- сплошной сбор с целевыми указаниями о конкретных ЭКБ. В этом случае в шаблон заполняются информация о конкретных изделиях ЭКБ.

2. Поисковый сбор предполагает работу по конкретным параметрам ЭКБ. Сбор происходит с помощью ввода определенных запросов. Например, «микросхема, K1556ХЛ8, наработка до отказа» и подобные.

Поисковый сбор используется в том случае, когда сплошной сбор избыточен. Нет смысла собирать информацию о всех параметрах ЭКБ изготовителя, нужен конкретный типонаминал и параметр.

Поисковый сбор часто используют по изготовителям, если необходимо отслеживать новые изделия. У изготовителя может быть тысячи типонаминалов ЭКБ, выполнять сплошной сбор всей категории, в которой представлены эти изделия, в данном случае нецелесообразно: дорого, долго и неэффективно. Достаточно отслеживать изделие по условному обозначению и по конкретному параметру.

3. Ссылочный сбор проходит по заранее подготовленным ссылкам на группы изделий или на отдельные изделия. Такой тип сбора используют для предприятий со специфической отраслевой особенностью, где алгоритмы автоматизированного и ручного сопоставления реализуются сложно в силу слабого соответствия отраслевой специфике. Например, СВЧ изделия ЭКБ. Подбирается ссылка, сопоставляя типонаминалы изделий в БД с типонаминалами у изготовителей и идентифицируются искомые позиции.

Также ссылочный сбор используется для сбора информации об ЭКБ небольших предприятий, у которых номенклатура не велика. Система формирует реестр ссылок с адресами и периодичностью опроса. Затем собираются данные по этим ссылкам. Система будет выполнять задание сбора, заходя по ссылке из реестра и собирая нужные параметры. Для получения актуальной информации ссылки требуют ручного обновления. Этот метод не подходит для большого объема данных, потому что это занимает много времени и трудоемко.

4. Сбор по параметрам ЭКБ. Предполагает сбор данных по параметрам из сопоставленной номенклатуры. Сбор по параметрам ЭКБ возможен только после реализации алгоритма с сопоставлением. Предварительно собирается информация сплошным сбором и автоматически сохраняются ссылки на информацию об изделиях. Затем происходит автоматизированное сопоставление данных изготовителя с данными в БД. Система накладывает матрицу номенклатуры изготовителя на номенклатуру в БД, сопоставляя их, и формирует реестр связанных данных.

Сбор по параметрам ЭКБ эффективен, когда необходимо ежедневно обновлять много данных, но нет необходимости каждый раз сплошным образом собирать целиком весь объем. При этом, можно выполнить обновление по ранее сопоставленным изделиям. Такая механика позволяет обновлять информацию только по нужным позициям. Многообразие типов сборов и понимание их различий позволяет подобрать индивидуальный ключ для эффективного решения по сбору данных для конкретных источников и типонаминалов ЭКБ.

При сборе больших данных использовать ссылочный сбор и мануальный подбор ссылок – бессмысленно, нерентабельно, неэффективно и даже опасно для динамического обновления.

В процессах исследований были обнаружены слабые места при обработке собранных данных. По результатам исправлены несколько важных недостатков с помощью процедуры ВРНД, разработанной в данном исследовании.



Анализ отношений и функциональных зависимостей

Для проведения ВРНД выполнен анализ отношений и функциональных зависимостей с использованием ER-моделей (Entity-Relationship model) [7]. Это формальная конструкция, описывающая концептуальные схемы взаимодействия элементов Системы, на этапе высокоуровневого их проектирования и инструментов управления ими, позволяющая выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

ER-модель позволяет получить конкретные схемы элементов и инструментов для модернизации БД. Метод достижения цели визуализации конкретных элементов БД и инструментов управления ими с помощью стандартной графической нотации визуализирования ER-модели – диаграммы сущность-связь (ER-диаграмма, entity-relationship diagram).

Элементы ER-модели БД Системы:

а) сущность ER-модели – объект, который идентифицируется способом, отличающим его от других объектов. Например, ЭКБ, функциональное назначение, параметр, показатель, модель.

б) набор сущностей – множество сущностей одного типа, обладающих одинаковыми свойствами, не исключаются пересекающиеся наборы. Например, номенклатура ЭКБ.

в) атрибут – элемент множества, описывающий свойства набора сущностей. Например, сущность – организация, атрибуты: разработчик, изготовитель, производитель, заказчик, калькодержатель, поставщик.

г) домен – множество значений (область определения) атрибута. Например, для атрибута наработка до отказа домен задается интервалом целых положительных чисел, поскольку наработка до отказа с отрицательным временем не бывает.

д) ключ сущности – один или более атрибутов уникально определяющих данную сущность. Например, условное обозначение ЭКБ.

е) связь – это ассоциация, установленная между несколькими сущностями. Пример, в микросхеме применяется корпус. Между сущностями корпус и микросхемы существует связь «применяется в». Но в данной микросхеме может применяться и другой корпус. Например, один для автоматизированного, второй для ручного монтажа.

Связь также может иметь атрибуты. Например, правила формирования выводов для конкретной микросхемы в конкретном корпусе.

ж) роль сущности в связи – функция, которую выполняет сущность в данной связи. Например, в связи корпус-микросхема сущности ЭКБ могут иметь роли «корпус» и «микросхема». Указание ролей в модели «сущность-связь» не является обязательным и служит для уточнения семантики связи.

з) набор связей – это связи между n (причем $n > 1$) сущностями, каждая из которых относится к некоторому набору сущностей. Например, гамма-процентная наработка до отказа изделия относится к показателям безотказности, гамма-процентный срок сохраняемости изделия относится к показателям сохраняемости, все они относятся к показателям надежности.

В случае $n=2$, когда связь объединяет две сущности, она называется бинарной. Доказано, что n -арный набор связей ($n > 2$) всегда можно заменить множеством бинарных, однако первые лучше отображают семантику взаимоотношений сущностей, что используется при построении ER-модели Системы.

То число сущностей, которое может быть ассоциировано через набор связей с другой сущностью, называют степенью связи. Рассмотрение степеней для бинарных связей используется при построении ER-модели Системы.

ER-диаграмма «один к одному» (1: 1). Например, условное обозначение изделия – микросхема, имеет обязательный класс принадлежности для обеих сущностей как показано на рисунке 1.





Рис. 1

Корпус-микросхема имеет необязательный класс принадлежности для обеих сущностей. Бескорпусная или одна микросхема может быть в различных корпусах как показано на ER-диаграмме рисунка 2.



Рис. 2

Один к многим (1: n). В данном случае сущности с одной ролью может соответствовать любое число сущностей с другой ролью. Такова связь, например, контакт-микросхема. В каждой микросхеме может быть разное число контактов, но контакт с ключом может принадлежать только одной микросхеме. Имеются особенности размещения ключей. Они могут быть разными в зависимости от типа корпуса. Например, для корпусов типа QFN, и зависеть от условного обозначения корпуса, как было исследовано ранее [8]. Графически степень связи n отображается «древовидной» линией, это показано на рисунке 3.

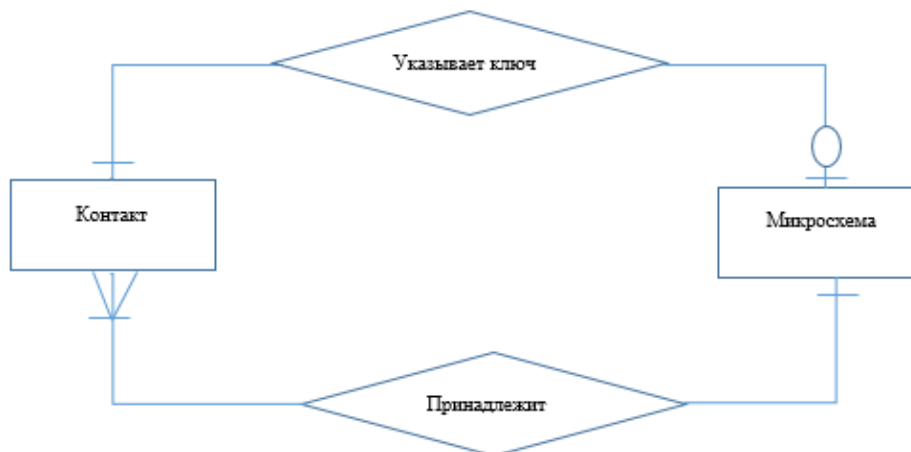


Рис. 3

Рисунок 3 дополнительно иллюстрирует тот факт, что между двумя сущностями может быть определено несколько наборов связей. Здесь также необходимо учитывать класс принадлежности сущностей. Каждый контакт должен принадлежать микросхеме, и микросхема должна иметь контакты. Поэтому сущность «Микросхема» имеет обязательный и сущность «Контакт» обязательный классы принадлежности.

Много к одному (n: 1). Эта связь аналогична отображению 1: n. Микросхема имеет группу параметров. Этот факт отображается в модели «сущность-связь» с помощью связи параметр-микросхема, объединяющей сущности параметр и микросхема. Так как микросхема имеет более одного параметра, то связь параметр – микросхема между этими сущностями будет иметь степень n: 1.

В условиях группы параметров или показателей (n: 1) используется метод классификации и маркировки информации [9]. Например, данные об отказах ЭКБ, которые являются важным параметром при их выборе для применения. ЭКБ, имеющая показатель надежности крайне низкий не может поменяться в РЭА ответственного назначения.



На рисунке 4 представляются атрибуты и набор атрибутов в виде прямоугольников, демонстрируется функциональная зависимость с помощью стрелок направления из входящих атрибутов.



Рис. 4

В данном случае, по совершенно очевидным соображениям каждый параметр принадлежит конкретной микросхеме, а каждая микросхема имеет параметры, каждая сущность имеет обязательный класс принадлежности.

Если существование одной сущности зависит от существования второй сущности, то первую называют зависимой сущностью. В качестве примера рассмотрим связь между сущностями «Пример применения» и «Микросхема». «Пример применения» появляется только после того, как будет обоснован и доказан для данной микросхемы, и прекращает свое существование при снятии микросхемы с производства. Таким образом, сущность «Пример применения» является зависимой от сущности «Микросхема». Зависимую сущность обозначают двойным прямоугольником, а ее связь с сильной сущностью линией, как показано на рисунке 5.



Рис. 5

Заметим, что кардинальность связи для сильной сущности всегда будет (1,1). Класс принадлежности и степень связи для зависимой сущности могут быть любыми. Предположим, например, что рассматриваемая микросхема имеет разрешение на применение в условиях отличных от указанных в документах на поставку, оформленный протоколом разрешения применения (ПРП). Указанное разрешение может распространяться на другие микросхемы с аналогичной группой параметров, применяемых в аналогичных условиях. Этот факт представляется набором сущностей ПРП и набором связей «распространяется на», как показано на рисунке 6. В случае снятия микросхемы с производства ПРП прекращает свое действие. Соответственно, возможность его распространения приостанавливается для данной схемы.



Рис. 6

Существует множество программных инструментов для автоматизированной работы с ER-моделями. Проведен сбор доступных и основные из них представлены в Таблице 1.



Таблица 1

Программные инструменты

Наименование программного инструментария для работы с ER-моделями	Лицензия
ARIS	Проприетарная
Avolution (англ.)	Проприетарная (EULA)
dbForge Studio for MySQL (англ.)	Проприетарная / бесплатное ПО (для некоммерческого использования)
Devgems Data Modeler (англ.)	Проприетарная
DeZign for Databases (англ.)	Проприетарная
Dia	Свободная (GNU GPL) [software 2]
ER/Studio (англ.)	Проприетарная
ERwin	Проприетарная
Fujaba	Свободная (GNU LGPL)
Innovator (нем.)	Проприетарная
MEGA International	Проприетарная
Microsoft Visio	Проприетарная
MySQL Workbench	Свободная (GNU GPL) / проприетарная (EULA)
OmniGraffle (англ.)	Проприетарная
Oracle Designer (англ.)	Проприетарная
PowerDesigner (англ.)	Проприетарная
Rational Rose (англ.)	Проприетарная
RISE Editor (англ.)	Проприетарная / бесплатное ПО [software 6]
SiSy (нем.)	Проприетарная
Sparx Enterprise Architect (англ.)	Проприетарная [software 7]
SQL Maestro	Проприетарная
SQLyog	Бесплатное ПО / проприетарная (EULA) [software 8]
StarUML (англ.)	Свободная (модифицированный вариант GNU GPL)
System Architect (англ.)	Проприетарная
Toad Data Modeler (англ.)	Проприетарная
Visual Paradigm (англ.)	Бесплатное ПО (для некоммерческого использования) [software 9]
yEd (англ.)	Бесплатное ПО (для некоммерческого использования) / yEd Software License Agreement



С точки зрения полноты функция доступности и разумной достаточности была выбрана уEd.

Была построена укрупненная модель, которая представлена в формате Hierarchical Layout – иерархический – это стиль компоновки, который описывает отношение приоритета ориентированных элементов модели. Он подходит для моделирования БД с целью нормализации данных и позволяет выделить основное направление или поток внутри модели [10]. Циклические зависимости элементов автоматически обнаруживаются и обеспечиваются связи. Элементы размещаются в иерархически расположенных слоях. Кроме того, порядок расположения элементов внутри каждого слоя выбирается таким образом, чтобы число пересечений элементов было минимальным.

ВРНД работает на среднем уровне абстракции. Выбор атрибутов для нормализации позволяет провести процедуры ВРНД и устранить слабые места в конструкции БД Системы.

В процедуре ВРНД используется метод постепенных изменений для лучшего выявления слабых мест и их устранения с помощью многократных построений ER-диаграмм. Цель состоит в том, чтобы с помощью методов процедуры ВРНД можно было достичь результатов оптимизации работы Системы. Эти методы процедуры ВРНД относятся к особому типу. Сначала рассматривается укрупнённая модель. Далее исследуются детали. Собираются элементы с помощью процедуры ВРНД и сравниваются с укрупнённой моделью.

Элементы модели были признаны необходимыми и позволяют администраторам иметь доступ к информации для организации проведения и контроля результатов нормализации.

Анализ процедуры валидации на уровне нормализации

При проведении исследований были выявлены слабые места БД и недостатки, которые были получены после реализации процедуры ВРНД, а затем разрешен вопрос: какие действия могут быть приняты на основе процедуры ВРНД, которые позволят устранить эти слабые места. Обнаружены три крайние точки в реализации процедуры ВРНД:

- а) проверка соответствия процедуры ВРНД на этапах первичного сбора данных об ЭКБ;
- б) идентификация значимых полей и записей, чтобы дополнить информацию от ЭКБ;
- в) корректировка информации об ЭКБ.

Первая крайняя точка имеет больше недостатков и слабых мест реализации. Так как к первичной информации относят информацию, отсутствие которой влечет за собой потерю дальнейших сведений об ЭКБ, описаний и нарушение логики восприятия информации, в которой выражена коммуникативная задача представления информации об ЭКБ. Следовательно, вторая и третья точки являются косвенными объектами для данного тематического исследования.

Первая крайняя точка требует полного соответствия ВРНД для правильного сбора первичной информации об ЭКБ.

Вторая точка нуждается в кастомизации. Дополнительная информация должна быть собрана для только предназначенных ЭКБ и для дополнения данных с использованием процедуры проверки на уровне нормализации. Эта задача также может быть решена в данных исследованиях.

Третья точка требует не только предназначенные для корректировки данные, но и проверку необходимости их корректировки.

Все действия на этом этапе, связанные с процедурой ВРНД выполнены и дали положительный результат:

- устранены проблемы с качеством информации об изделиях ЭКБ, возникающие при импорте данных из внешних источников;
- данные приведены к единым масштабам единиц измерения – для одних и тех же параметров, показателей, других характеристик;
- используемая память уменьшилась, а эффективность обновления информации в БД повысилась также как повысился показатель целостности данных.



Заключение

Процедура ВРНД играет ключевую роль в процессах организации БД и сопровождения ее нормального функционирования. В проведенных исследованиях процедура была разделена на различные этапы в зависимости от способа сбора информации.

При этом получены положительные результаты ВРНД, подтверждающие нормализацию данных. Дефицит данных и повторение значений нарушает нормализацию и создает проблемы с выбором ЭКБ для применения в РЭА. Большой проблемой также являются неточные форматы данных.

Положительные значения ВРНД получены с использованием активного набора NF в терминах классов эквивалентности атрибутов, на разных уровнях записи, таких как 1NF, 2NF, 3NF, нормальной формы Бойса-Кодда, 4NF, пятой нормальной формы 5NF. Апробирована, но не дала ожидаемых результатов нормализации форма проекции-соединения PJ/NF.

Разработана открытая анкета, предназначенная для сбора позиций источников информации о структуре шаблона. В результате анализа полученных анкет, сформированы шаблоны и определены типы сбора информации об ЭКБ.

На основании шаблонов и проработанных типов сбора информации выполнен анализ отношений и функциональных зависимостей с использованием ER-моделей. Определен инструмент разработки ER-моделей. Разработана укрупненная модель данных в формате Hierarchical Layout.

Выполнена процедура ВРНД. Результаты положительные и описаны в предыдущем разделе. При этом определено: каждый вид входной информации об ЭКБ имеет свои особенности в части ВРНД. ЭКБ совершенствуется. Поэтому имеется еще много возможностей для исследований с использованием описанного выше метода.

Список литературы:

1. Philipp Skavantzios, Sebastian Link Third and Boyce–Codd normal form for property graphs // The VLDB Journal, New Zealand. 2025. №34 (2). P.1-26
2. Millist W. Vincent A corrected 5NF definition for relational database design // Theoretical Computer Science, South Australia. 1997. №185. P.379-391.
3. Горнев Е.С. Тельминов О.А. Отечественная элементная база и программное обеспечение для реализации нейросетевых решений // III-я Международная конференция «Математическое моделирование в материаловедении электронных компонентов». Пленарный доклад 25.10.2021 г.
4. Рубцов Ю.В. Оптимизация процессов подбора аналогов изделий электронной компонентной базы // Электронная техника. Серия 3: Микроэлектроника. 2015. С. 25-30.
5. Aaron S. PHP vs. JavaScript: A Thorough Comparison // BigDegree – 2025 – URL: <https://www.bitdegree.org/tutorials/php-vs-javascript> (дата обращения: 28.08.2025 г.)
6. Шмаков С.Э. Проектирование приложения для сбора данных из сторонних интернет-источников // Электронный научный журнал «Инженерный вестник Дона». 2023. № 12.
7. Antonio Badia. Entity-Relationship modeling revisited // ACM SIGMOD Record University of Louisville. 2004. №33 (1). P. 77-82.
8. Гагарина Л. Г., Рубцов Ю.В. Особенности разработки метода классификации плоских QFN-корпусов для применения в составе автоматизированных систем технической подготовки производства изделий микроэлектроники // Известия высших учебных заведений. Электроника. Т. 27. 2022. №3. С. 322-332.
9. Дормидошина Д.А., Савин М.Л., Рубцов Ю.В. Применение ИСМН в процессах сбора, обработки и анализа информации о надежности изделий микроэлектроники // «Нано- и микросистемная техника». 2020. №9. С.485-488.
10. Nikola S. Nikolov, Patrick Healy. Hierarchical Drawing Algorithms // In book: Handbook of Graph Drawing and Visualization University of Limerick. Ireland. 2013. №1. P. 409-454.

