

ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ НА УРОВНЕ БАЗЫ ДАННЫХ

О. А. Молокович¹, П. Е. Дадонов²

¹ o.molokovich@ya.ru, ² ozf_dad@mail.ru

ФГБОУ ВО «Уфимский государственный авиационный технический университет» (УГАТУ)

Аннотация. Рассматривается один из подходов к интеграции информационных систем на платформе Oracle 11g и обеспечения непротиворечивости данных. Предложен алгоритм интеграции на уровне базы данных.

Ключевые слова: интеграция информационных систем; база данных; унаследованные системы; Oracle 11g.

ВВЕДЕНИЕ

В современном мире едва ли найдется сфера жизни общества, в которой не применяются информационные технологии, особенно важное место они занимают в промышленности, банковском деле и медицине. Во всех этих областях существуют различные информационные системы, использующие реляционные базы данных и общие данные, хранящиеся в них, например, общероссийские классификаторы по разным категориям (налоговые инспекции, коды причин постановки на учет, единицы измерения и другие). Во многих организациях встает вопрос совместной работы этих систем и непротиворечивости общеиспользуемых данных.

Этой проблемой занималось немало ученых, таких как И. С. Михайлова, Д. В. Торшин, В. В. Вейбер и другие. В данной статье также предложен один из подходов к решению этой проблемы на примере базы данных под управлением Oracle 11g версии 11.2.0.4. При этом все подсистемы реализованы на одной технологической платформе.

ОПИСАНИЕ ЗАДАЧИ

Рассмотрим множество информационных подсистем, хранящих и использующих данные общероссийских классификаторов. Эти подсистемы разрабатывались разными проектными командами, в разное время и имеют различный формат представления

данных, поэтому одна и та же информация в справочниках этих подсистем представлена по-разному.

Требуется организовать совместную работу этих подсистем и решить проблему противоречивости и дублирования информации в справочниках. Было принято решение организовать единое справочное пространство для хранения и совместного использования данных классификаторов и общих сущностей, таких как пользователи и аудит их действий.

В связи с этим возникает вопрос, каким образом обеспечить удобный и наименее затратный с финансовой точки зрения способ работы с этими данными, требующий минимальных изменений в существующем программном обеспечении.

ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Для начала предлагается выделить множество общих справочников и сущностей, используемых многими системами организации. Это могут быть как общероссийские классификаторы, так и справочники внутри организации, например, список пользователей систем или организационная структура предприятия.

Требуется тщательно продумать структуру таблиц, особенно, для общих справочников, загружаемых автоматически. Крайне не рекомендуются поля со свободным заполнением, поскольку один и тот же объект можно описать по-разному, а с точки зрения системы это будут разные записи. Значения

данных столбцов таблиц либо должны быть даны в источнике, либо вычисляемыми по определенными правилам. Для успешной интеграции подсистем предприятия следует использовать данные только из официальных источников, например, для справочника налоговых инспекций код и название прописаны в системе обозначений налоговых органов, предоставляемой акционерным обществом «Главный научный инновационный внедренческий центр».

Поскольку в реализации существующего программного обеспечения могут быть ссылки на таблицы справочников, которые предполагается вынести в общее пространство, необходимо особое внимание уделить полям первичного и суррогатного первичного ключей, их типам данных и размерности. Например, в подсистеме А справочник кодов налоговых органов имеет суррогатный первичный ключ типа NUMBER(19), в подсистеме Б роль первичного ключа выполняет код инспекции типа CHAR(4). Следовательно, типы полей внешних ключей дочерних таблиц отличаются. Выходом из этой ситуации может быть создание суррогатного первичного ключа NUMBER(19) и уникального ограничения на код налогового органа CHAR(4). СУБД Oracle 11g позволит создать ограничения ссылочной целостности на оба поля, что позволит не переписывать таблицы и код приложения подсистемы Б.

Затем необходимо создать нового пользователя базы данных и в его схеме разместить следующие объекты [1]:

- таблицы для хранения общих данных;
- представления, возвращающие все или часть данных, доступных той или иной подсистеме;
- последовательности для заполнения значений суррогатного первичного ключа;
- триггеры, например, для заполнения вычисляемых полей.

После того, как все необходимые объекты пользователя общего информационного пространства будут созданы следует дать права на [2]:

- создание ссылок на таблицы (grant references);
- запрос данных из представлений (grant select).

Следующим шагом следует составить таблицы соответствия записей из разных источников. Для этого требуется определить поле или группу полей справочника, идентифицирующие записи однозначно. Сопоставить записи таблиц справочника всех интегрируемых подсистем и сохранить данные трансляции в буферную таблицу. Затем следует заполнить таблицы общего пространства значениями справочника из всех подсистем и дополнить таблицу трансляции идентификаторами созданных записей. После этого необходимо обновить значения внешних ключей в дочерних таблицах интегрируемых подсистем. Для обеспечения целостности данных следует переключить внешние ключи таблиц интегрируемых систем на таблицы общего пространства [4]. Исходные таблицы удалить (или переименовать).

Затем в схемах пользователей интегрируемых подсистем следует создать [1]:

- синонимы на представления пользователя единого информационного пространства;
- представления на выборку данных из представления пользователя общего пространства через созданные синонимы;
- процедуры и функции;
- синонимы на описанные в предыдущих пунктах представления, процедуры и функции для обращения к ним из приложения, если требуется.

Возможно, читателю покажется весьма странной рекомендация создать представление в схеме подсистемы на представление справочника в схеме общего пространства, но именно такое сочетание объектов обеспечивает гибкость предлагаемого решения, поскольку позволяет вносить изменения в структуру хранения в общем пространстве без опасений, что это скажется на зависимых подсистемах. Также с помощью представлений можно разграничить доступ к данным общего пространства, например, фильтровать список пользователей. Представление в схеме пользователя подсистемы позволяет форматировать данные, управлять их типами и варьировать состав и наименования полей независимо от общего пространства. Возможность создания собственного представления в каждой интегрируемой подсистеме смягчает требование к

составу полей в справочниках единого пространства. В ситуации, когда список полей справочника в интегрируемой подсистеме по каким-то причинам был шире, чем указано в официальном источнике, расширенные поля используются только этой подсистемой и заполнить их автоматически не представляется возможным, можно вынести их в отдельную таблицу вне единого пространства. Таблица с дополнительными полями будет дочерней для таблицы справочника в общем пространстве, а представление поможет собрать их вместе, таким образом клиентское приложение «не заметит» изменения в структуре хранения данных.

Схема расположения объектов базы данных представлена на рис. 1.

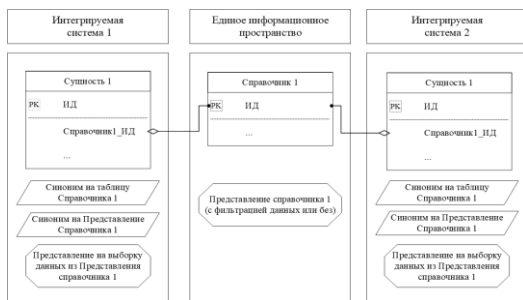


Рис. 1. Схема расположения объектов базы данных

ПРИМЕР

Рассмотрим процесс миграции справочника в общее пространство на примере «Коды налоговых инспекций» (КНО). Структура таблицы представлена на рис. 2.

REF_KNO			
CODE	CHAR(4)	<pk>	not null
ADDRESS	VARCHAR2(255 char)		null
NAME	VARCHAR2(512 char)		not null
SHORT_NAME	VARCHAR2(255 char)		not null
START_DATE	DATE		not null
END_DATE	DATE		null
LAST_MODIFIED_DATE	TIMESTAMP		not null
LAST_MODIFIED_BY	NUMBER(19)		not null
COMMENT_TEXT	VARCHAR2(1023 char)		null
REASON	NUMBER(19)		not null
INN	CHAR(10)		null
KPP	CHAR(9)		null
PHONES	VARCHAR2(100 char)		null
EMAIL	VARCHAR2(64 char)		null
ADDRESS_ID	NUMBER(19)		null
Коды налоговых инспекций			

Рис. 2. Структура справочника КНО в интегрируемой подсистеме

В качестве первичного выступает поле «Код» типа CHAR(4). В других подсистемах, использующих этот справочник, пер-

вичным ключом является поле ID типа NUMBER(19). Было принято решение использовать второй вариант.

Алгоритм миграции следующий:

1. Создать таблицу справочника в общем пространстве. Структура представлена на рисунке 3. Поле ID – суррогатный первичный ключ, на поле CODE установлено уникальное ограничение.

2.

REF_TAX_ORG_CODE			
CODE	CHAR(4 char)	<pk2>	not null
NAME	VARCHAR2(512 char)		not null
SHORT_NAME	VARCHAR2(255 char)		null
ADDRESS	VARCHAR2(255 char)		null
INN	CHAR(10 char)		null
KPP	CHAR(9 char)		null
PHONES	VARCHAR2(100 char)		null
EMAIL	VARCHAR2(64 char)		null
ADDRESS_ID	NUMBER(19)		null
START_DATE	DATE		not null
END_DATE	DATE		null
COMMENT_TEXT	VARCHAR2(1023 char)		null
LAST_MODIFIED_DATE	TIMESTAMP		not null
LAST_MODIFIED_BY	NUMBER(19)		not null
REASON	NUMBER(19)		not null
ID	NUMBER(19)	<pk,akt2>	not null
Коды налоговых инспекций			

Рис. 3. Структура справочника КНО в едином пространстве

3. Создать представление VW_REF_TAX_ORG_CODE, возвращающее все записи справочника с перечислением всех полей, в схеме общего пространства.

4. Создать последовательность REF_TAX_ORG_CODE_SEQ для заполнения суррогатного первичного ключа в таблице REF_TAX_ORG_CODE.

5. Скопировать данные из REF_KNO в REF_TAX_ORG_CODE, рекомендуется с помощью MERGE, используя REF_KNO.CODE в качестве условия сопоставления, если запись еще не создана в REF_TAX_ORG_CODE, вставить ее, используя следующее значение последовательности REF_TAX_ORG_CODE_SEQ для заполнения REF_TAX_ORG_CODE.ID.

6. Выдать права grant references пользователю рассматриваемой подсистемы на таблицу REF_TAX_ORG_CODE справочника КНО в общем пространстве и чтение данных из представления VW_REF_TAX_ORG_CODE.

7. Создать синонимы в схеме интегрируемой подсистемы REF_TAX_ORG_CODE_SYN на таблицу REF_TAX_ORG_CODE и

VW_REF_TAX_ORG_CODE_SYN на представление VW_REF_TAX_ORG_CODE в схеме единого пространства.

8. Переименовать таблицу REF_KNO справочника КНО в рассматриваемой подсистеме на REF_KNO_OLD.

9. В схеме рассматриваемой подсистемы создать представление REF_KNO, имя которого совпадает с наименованием исходной таблицы.

10. Пересоздать внешние ключи на справочник, используя в качестве родительской таблицы REF_TAX_ORG_CODE. Это можно сделать с помощью системного представления ALL_CONSTRAINTS, запросив все ограничения целостности, ссылающиеся на REF_KNO_OLD.

11. Удалить таблицу REF_KNO_OLD.

12. Закрывать доступ на редактирование справочника КНО в интегрируемой ФП.

13. Организовать редактирование справочника из интерфейса для работы с общим пространством или организовать актуализацию данных с помощью аппарата Extract, Transform, Load (ETL).

Если все же редактирование из интерфейса интегрируемой подсистемы закрыть невозможно, то следует расширить права на объекты схемы единого пространства пользователю интегрируемой подсистемы.

Рассмотренный процесс миграции показывает пример сокрытия структуры хранения данных и гибкость решения.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Рассматриваемый подход в конечной реализации предполагает наличие отдельного интерфейса для редактирования и добавления записей в общие таблицы. Такая точка входа для изменения данных обеспечивает их актуальность и непротиворечивость в интегрируемых системах. Удаление записей из таблиц разделяемого информационного пространства исключается (или ограничивается) в виду вероятности существования записей в дочерних таблицах. Редактирование в интегрируемых системах закрывается, но возможны модификации, при которых данные могут изменяться не только из интерфейса единого информационного пространства. Вариант, когда отдельный интерфейс

отсутствует тоже имеет место на этапе внедрения, тогда потребуется изменить набор прав пользователей подсистем. Но такие решения должен принимать архитектор (аналитик), знающий все нюансы предметной области, специфики работы организации и ее структуры.

Разработанная комбинация из таблиц, представлений, синонимов и наборов прав позволяет обеспечить безопасность доступа к данным, сокрытие истинной структуры их хранения и представление данных для каждой системы в необходимом ей виде, что позволяет не переписывать существующие приложения и сократить расходы на сопровождение программного обеспечения.

ЗАКЛЮЧЕНИЕ

Предложенный подход был успешно применен авторами во время интеграции информационных систем одного из крупнейших предприятий на территории Российской Федерации и функционирует в промышленной эксплуатации.

СПИСОК ЛИТЕРАТУРЫ

1. **Официальная документация** СУБД Oracle «Database Concepts» [Электронный ресурс]. — Режим доступа: URL: https://docs.oracle.com/cd/E11882_01/server.112/e40540/d_atainte.htm#CNCPT021 (30.08.2018). [Oracle database official documentation «Database Concepts» [Online]. Available: https://docs.oracle.com/cd/E11882_01/server.112/e40540/d_atainte.htm#CNCPT021]
2. **Официальная документация** по СУБД Oracle, Привилегии [Электронный ресурс]. — Режим доступа: URL: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9013.htm (30.08.2018).). [Oracle database official documentation, Grant. [Online]. Available: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9013.htm]
3. **Кузнецов С. Д.** Проектирование и разработка корпоративных информационных систем — Центр Информационных Технологий, 1998 [Электронный ресурс]. — Режим доступа: URL: http://citforum.ru/cfin/prcorpsys/infstpr_13.shtml (10.02.2018).). [S. D. Kuznetsov, 2018, Feb. 10). Design and development of corporate information systems". In Centr Informatsionnyh Tehnologiy, 1998 [Online], (in Russian). Available: http://citforum.ru/cfin/prcorpsys/infstpr_13.shtml]
4. **Официальная документация** по СУБД Oracle, Ограничения [Электронный ресурс]. — Режим доступа: URL: https://docs.oracle.com/cd/B19306_01/server.102/b14200/causes002.htm (30.08.2018).). [Oracle database official documentation, Constraint [Online]. Available: https://docs.oracle.com/cd/B19306_01/server.102/b14200/causes002.htm]

5. **Официальная** документация по СУБД Oracle, Представления [Электронный ресурс]. — Режим доступа: URL: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8004.htm (30.08.2018).). [Oracle database official documentation, View [Online]. Available: https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8004.htm]

ОБ АВТОРАХ

МОЛОКОВИЧ Ольга Александровна, магистрант, каф. ВМиК.

ДАДОНОВ Павел Евгеньевич, студент каф. ВМиК

METADATA

Title: Database level information systems integration

Authors: O. A. Molokovich¹, P. E. Dadonov²

Affiliation:

Ufa State Aviation Technical University (UGATU), Russia.

Email: ¹o.molokovich@ya.ru, ²ozf_dad@mail.ru

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), no. 1 (20), pp. 114-118, 2019. ISSN 2225-9309 (Print).

Abstract: Discusses information system on the Oracle 11g platform integrating and data consistency problem. A database integration algorithm is proposed.

Key words: integration of information systems; database; legacy systems; Oracle 11g.

About authors:

MOLOKOVICH, Olga Alexandrovna, master student 1 year, Ufa state aviation technical University

DADONOV, Pavel Yevgenyevich, bachelor student 1 year, Ufa state aviation technical University