

РУКОВОДСТВО ПО МИГРАЦИИ С ORACLE НА SYBASE ASE



Содержание

1	Введение	4
1.1	Целевая аудитория	4
1.2	Вводные сведения	4
1.3	О сервере Sybase ASE	4
1.4	Системы Oracle, рассматриваемые в данном Руководстве	4
1.5	Сравнение продуктов Oracle и Sybase	4
1.6	Версии баз данных Oracle и Sybase, к которым применимо данное Руководство	5
1.7	Sybase ASE – документация и справочные данные	5
2	Как пользоваться данным Руководством	6
2.1	Общее представление о миграции	6
2.2	Факторы успеха	6
2.3	Вопросы, не рассматриваемые в данном Руководстве: различные аспекты проекта	7
2.4	Вопросы, не рассматриваемые в данном Руководстве: тонкая настройка Sybase ASE	7
3	Оценка сложности миграции	8
3.1	Контрольный перечень Oracle: типы данных	8
3.2	Контрольный перечень Oracle: категория «Простая конверсия»	8
3.3	Контрольный перечень Oracle: категория «Частичное переписывание»	11
3.4	Контрольный перечень Oracle: категория «Полное переписывание»	13
4	Миграция схемы базы данных	15
4.1	Определение схемы Oracle	16
4.1.1	Использование существующих скриптов DDL	16
4.1.2	Декомпилирование существующей схемы	16
4.2	Использование Sybase PowerDesigner для миграции схемы базы данных	16
4.2.1	Этапы конвертирования схемы с помощью PowerDesigner	16
4.3	Декомпилирование схемы Oracle без использования PowerDesigner	17
4.4	Особые случаи миграции схем	17
4.5	Сопоставление схемы Oracle с базами данных Sybase ASE	18
4.6	Терминология Oracle–Sybase, связанная со схемами	18
4.7	Сопоставление типов данных Oracle и Sybase ASE	19
4.8	Поиск зарезервированных слов в Sybase ASE и ключевых слов в Oracle	22
4.9	Выбор схемы блокировки для таблиц в Sybase ASE	22
4.10	Таблица DUAL в Oracle	23
5	Миграция аспектов серверного уровня	24
5.1	Набор символов	24
5.2	Чувствительность к регистру сервера баз данных (порядок сортировки)	24
5.3	Параметры конфигурации сервера	24
5.4	Дисковая подсистема	25
5.5	Миграция логинов пользователей	25
5.5.1	Пароли пользователей	25
5.6	Права доступа	26
6	Миграция данных	27
6.1	Выгрузка данных Oracle в ASCII-файлы; загрузка в ASE с помощью утилиты bcp	27
6.1.1	Загрузка в ASE с помощью bcp	27
6.1.2	Выгрузка из Oracle: FACT (инструмент стороннего производителя)	28
6.1.3	Выгрузка из Oracle: разверните собственную утилиту PL/SQL для экспорта данных Oraclета	28
6.2	Использование технологии Sybase Enterprise Connect Data Access (ECDA) для Oracle	29
6.2.1	Пример использования ECDA	29
6.3	Использование Sybase Replication Server Heterogeneous Edition (RSHE) для Oracle	30
6.3.1	Минимальное время простоя при миграции с использованием сервера репликации	30
6.3.2	Исходная материализация для настройки репликации	30
6.3.3	Прочие соображения	31

6.4	Использование инструмента ETL стороннего разработчика, которое поддерживает и Oracle, и Sybase ASE.	31
6.5	Типы данных Oracle, требующие особого внимания при миграции	31
7	Миграция PL/SQL в Transact-SQL	33
7.1	Местонахождение кода PL/SQL	33
7.2	SwisSQL – помощь при миграции PL/SQL в T-SQL	33
8	Сравнение транзакций и блокировки в Oracle и Sybase	35
8.1	Сравнение блокировки в Oracle MVCC и Sybase	35
8.2	Вопросы миграции, связанные с транзакциями	35
8.3	Использование в ASE неявных/связанных транзакций	36
8.3.1	Язык управления транзакциями DDL	36
8.3.2	Обработка транзакций в сохраненных процедурах	36
8.4	Использование в ASE режима явных/несвязанных транзакций	36
8.5	Использование в ASE усовершенствований, обеспечивающих параллельное выполнение транзакций	36
8.6	Другие аспекты выполнения транзакций	37
9	Прочие аспекты миграции	39
9.1	Курсоры	39
9.2	Последовательности	39
9.3	Обработка ошибок/исключений	40
9.4	Ограничения на внешнее присоединение	40
9.5	Миграция приложений JDBC/ODBC/...	41
9.5.1	JDBC	41
9.6	Формы Oracle	41
10	Перекрестные ссылки для задач администрирования баз данных (DBA)	42
11	Перекрестные ссылки при миграции из Oracle на Sybase	46
11.1	Миграция из Oracle на Sybase ASE: категория «Простая конверсия»	46
11.2	Миграция из Oracle на Sybase ASE: категория «Частичное переписывание»	55
11.3	Миграция из Oracle на Sybase ASE: категория «Полное переписывание»	61

История изменений:
 Редакция 1.0, сентябрь 2011 г.

© 2011 Sybase, Inc.

Sybase, Transact-SQL, Adaptive Server Enterprise и Replication Server являются зарегистрированными товарными знаками компании Sybase, Inc.

Названия других продуктов или другие фирменные наименования могут являться (зарегистрированными) товарными знаками соответствующих владельцев.

1 Введение

Цель настоящего Руководства по миграции (далее – Руководство) – предоставить инструкции и помочь в процессе перехода с базы данных Oracle на Sybase ASE (Adaptive Server Enterprise). Под миграцией понимают такое изменение клиент-серверного приложения, использующего в настоящий момент базу данных Oracle в качестве СУБД, в результате которого оно перейдет на использование базы данных Sybase ASE.

В данном Руководстве главное внимание уделяется переносу основных функций Oracle на Sybase ASE. При этом не рассматриваются аспекты работы Sybase ASE, связанные с производительностью (см. также раздел 2.4).

1.1 Целевая аудитория

Настоящее Руководство предназначено для специалистов, которые занимаются миграцией баз данных Oracle на сервер Sybase Adaptive Server Enterprise (ASE).

1.2 Вводные сведения

Предполагается, что читатель уже знаком с реляционными базами данных, в частности с БД Oracle. Кроме этого, требуются знания начального уровня в области СУБД Sybase ASE.

Чтобы осуществить успешную миграцию базы данных, специалист должен отлично разбираться в СУБД Oracle, в том числе в архитектуре верхнего и нижнего уровня этих СУБД, а также во взаимодействии клиентских приложений и базы данных Oracle.

1.3 О сервере Sybase ASE

Sybase ASE – это СУБД, на которой работает большая часть компаний Уолл-Стрит. За последние 25 лет сервер ASE обеспечивал исключительную надежность и высочайшее быстродействие. Совокупная стоимость владения сервером Sybase ASE ниже по сравнению с Oracle, а производительность при использовании одинаковой аппаратной платформы – выше. Sybase ASE подходит для размещения баз данных любого приложения, которое в настоящее время работает под управлением Oracle.

1.4 Системы Oracle, рассматриваемые в данном Руководстве

Данное Руководство можно использовать в процессе миграции любых систем на основе СУБД Oracle. Приведенные в нем сведения не ограничиваются определенным типом приложения, рабочей нагрузкой или архитектурой системы, однако предполагается, что большинство систем на основе СУБД Oracle – кандидатов для миграции – используются для выполнения транзакций.

Особо отметим, что в данном Руководстве специально не рассматриваются вопросы перехода на Sybase ASE систем SAP Business Suite, работающих под управлением Oracle. Для миграции этих систем компания SAP предлагает соответствующие продукты и услуги, и клиенты, для которых этот вопрос актуален, должны обращаться непосредственно в SAP.

1.5 Сравнение продуктов Oracle и Sybase

Компании Oracle и Sybase предлагают множество продуктов для работы с базами данных. В списке ниже приведено сопоставление основных продуктов Oracle верхнего уровня с соответствующими продуктами Sybase. Несмотря на то, что этот список мы специально сделали кратким, им можно руководствоваться для поиска аналогичных продуктов этих компаний.

Данное Руководство посвящено миграции данных с сервера баз данных Oracle Database Server на сервер Sybase ASE. Обычно считается, что данные системы – OLTP-ориентированные, однако это не является обязательным требованием.

Oracle	Sybase
Oracle Database Server	Sybase ASE (Adaptive Server Enterprise)
Oracle OLAP и DW	Sybase IQ
Oracle Analytics	Sybase IQ
Oracle RAC	Sybase ASE Cluster Edition
Oracle Times Ten	Sybase ASE 15.5 In-Memory Database
Oracle Streams	Sybase Replication Server

1.6 Версии баз данных Oracle и Sybase, к которым применимо данное Руководство

Настоящий документ распространяется на версии Oracle 9i, 10g и 11g.

Предполагается, что миграция осуществляется на сервер Sybase ASE версии 15.5 (или более поздней). Не рекомендуется выполнять миграцию на более ранние версии ASE, поскольку этот процесс не рассматривается в данном Руководстве.

Если не указано иное, то все ссылки на ASE или Adaptive Server считаются ссылками на Sybase Adaptive Server Enterprise.

1.7 Sybase ASE – документация и справочные данные

Для получения более подробных сведений о Sybase ASE см. <http://www.sybase.com/ase>, где содержатся документы общего характера и подробные отчеты.

Документацию для ASE и руководства по использованию отдельных продуктов можно найти на сайте <http://infocenter.sybase.com>. Особого внимания заслуживают следующие документы, относящиеся к серверу ASE:

- Руководство пользователя по Transact SQL
- Справочное руководство
- Руководство по администрированию системы
- Руководство по использованию сервисных программ
- Руководство по быстрой настройке и настройке

Помимо этого, компания Sybase проводит техническое обучение работе с сервером ASE. Подробное описание имеющихся курсов приведено здесь:

<http://www.sybase.com/education>.

2 Как пользоваться данным Руководством

В Руководстве главное внимание уделяется техническим аспектам, характерным для конкретной базы данных при миграции БД Oracle на БД Sybase. При анализе работ, необходимых для реализации проекта миграции, Руководство поможет оценить его сложность, вовремя выявить различные аспекты, которые могут затруднить переход. Кроме того, Руководство поможет разработать подход к миграции, поскольку предлагает подробные технические решения различных проблем, связанных с этим процессом.

2.1 Общее представление о миграции

Мы рекомендуем осуществлять миграцию с Oracle на Sybase ASE поэтапно. В порядке важности и приоритетности можно выделить следующие этапы:

1. Оценка сложности миграции при помощи контрольного перечня, приведенного в главе 3. На данном этапе выявляются специфические характеристики имеющихся баз данных Oracle, для которых могут отсутствовать эквивалентные опции в Sybase.

Настоятельно рекомендуется уделить данному вопросу достаточно внимания, чтобы обозначить и правильно оценить наиболее сложные моменты перехода.

2. Миграция схемы базы данных – первый шаг процесса перехода (описан в главе 4). На этом этапе мы рекомендуем использовать инструмент Sybase PowerDesigner, который позволит получить схему Oracle и конвертировать ее в эквивалентную схему Sybase ASE.
3. Миграция объектов серверного уровня, например, таких как «пользователи» (описана в главе 5).
4. Миграция данных (описана в главе 6). При выполнении миграции данных следует руководствоваться подходом, предполагающим максимально приемлемое время простоя для приложения.

Рекомендуем рассмотреть возможность применения инструментов сторонних производителей для извлечения данных из системы Oracle. Если критически важным является минимальное время простоя приложения, то следует подумать об использовании сервера репликации Sybase Replication Server – это позволит свести время простоя к нескольким минутам вместо часов.

5. Миграция кода Oracle PL/SQL в Sybase Transact-SQL (см. также главу 7). Ее необходимо выполнить для кода SQL базы данных (например, для сохраненных процедур, триггеров, SQL-функций), а также для кода клиентских приложений. Как правило, этот этап представляет собой наиболее сложную часть процесса миграции. Рекомендуем оценить возможность использования инструмента SwissQL от стороннего производителя (см. раздел 7.2), который поможет в ходе миграции PL/SQL.

Чтобы облегчить данный этап миграции, обратитесь к главе 11. В ней содержатся перекрестные ссылки между характеристиками Oracle и их эквивалентами в Sybase ASE, разбитые на три категории: «Простая конверсия», «Частичное переписывание» и «Полное переписывание». Эти ссылки представляют собой расширенный вариант контрольного перечня Oracle, приведенного в главе 3.

6. Миграция инфраструктурных компонентов, зависящих от разработчика, например JDBC-драйверов (см. раздел 9.5).
7. Конвертация задач техобслуживания, администрирования и мониторинга. Все указанные аспекты очень сильно отличаются для БД разных производителей, поэтому здесь не совсем правильно использовать термин «миграция». В главе 10 приводятся перекрестные ссылки для некоторых распространенных аспектов администрирования баз данных (DBA). Однако этого недостаточно для выполнения миграции – потребуется квалификация специалиста в области DBA применительно к продуктам Oracle и Sybase.
8. Главное внимание в данном Руководстве уделено вопросам достижения функциональной эквивалентности исходной системы на Oracle и конечной системы на Sybase ASE. На следующем шаге, скорее всего, потребуется выполнить оптимизацию и настройку с учетом особенностей Sybase ASE, чтобы обеспечить необходимый уровень производительности. Настройка, учитывающая особенности Sybase ASE, не рассматривается в данном Руководстве (см. раздел 2.4).

2.2 Факторы успеха

Миграция базы данных – довольно сложная процедура, поэтому необходимо сделать все, чтобы избежать дорогостоящих сбоев в этом процессе. Ниже перечислены ключевые факторы успеха, которые применимы к любому проекту миграции баз данных из Oracle на Sybase:

- Знание предметной области соответствующих бизнес-приложений, системы и среды. Крайне важно иметь полное и целостное представление о работе всех приложений, обращающихся к базе данных Oracle. Сюда входят клиентские приложения, которые подключаются к базе данных Oracle как непосредственно, так и косвенно, например, через сервер приложений.
Необходимо понимать, к каким данным в БД имеет доступ каждое из этих приложений и как оно модифицирует эти данные. Следует выявить каждый код SQL, передаваемый приложением в базу данных, а также способы его изменения.
- В данном случае абсолютным требованием является наличие достаточной квалификации для работы с продуктами Oracle, которое позволит проанализировать все аспекты базы данных. Самое главное – выявить специфические особенности Oracle (согласно контрольным перечням, приведенным в главе 3), в особенности те, которые не имеют прямого эквивалента в Sybase.
- При этом используется прямой доступ к коду Oracle PL/SQL – как в базе данных, так и во всех клиентских приложениях.
- Как минимум требуется достаточное понимание принципов построения Sybase ASE, чтобы создать функционально работающую систему с перенесенной базой данных. На последующих этапах миграции необходимо углубленное знание продуктов Sybase, чтобы настроить и оптимизировать работу с учетом особенностей Sybase ASE. На начальных стадиях проекта весьма полезной будет помощь эксперта соответствующей квалификации.
- Необходимо провести всестороннее тестирование в среде, аналогичной реальной рабочей среде, чтобы проверить правильность подхода, выбранного для проведения миграции, и определения программных приложений, затрагиваемых этим процессом. Для получения наилучших результатов настоятельно рекомендуем использовать копию реальных производственных данных (или как можно более близких к ним), а также аппаратное обеспечение, аналогичное тому, которое используется на производстве.

2.3 Вопросы, не рассматриваемые в данном Руководстве: различные аспекты проекта

Данное Руководство не содержит никаких предписаний или предложений по организации проекта миграции с точки зрения подготовки, задания процедур тестирования, проверки перенесенных компонентов и так далее. Все эти аспекты оставлены на усмотрение самой организации, в соответствии с установленными требованиями и стандартами, имеющимися передовыми практиками и предпочтениями. Следует иметь в виду, что отсутствие специальных рекомендаций по тестированию и проверке перенесенных компонентов не означает, что данные действия не нужно выполнять. Напротив, они очень важны, и в этом вопросе рекомендуется придерживаться общепринятых современных практик применительно к тестированию и проверке правильности работы программного обеспечения.

2.4 Вопросы, не рассматриваемые в данном Руководстве: тонкая настройка Sybase ASE

Главная цель данного Руководства – помочь в создании системы на основе Sybase ASE, которая будет функционально эквивалентна исходной системе на основе СУБД Oracle. Целью данного Руководства не является предоставление инструкций, следуя которым можно получить оптимально настроенную систему Sybase ASE. Несмотря на то, что в ходе миграции потребуется настройка с учетом особенностей Sybase ASE, все вопросы, связанные с настройкой, умышленно вынесены за рамки этого Руководства.

Поскольку считается, что тонкая настройка сервера ASE с учетом его специфики совершенно не связана с какими бы то ни было аспектами или соображениями, касающимися продуктов Oracle, мы отсылаем читателя к документации по Sybase ASE. В ней можно почерпнуть основные сведения и рекомендации по настройке Sybase ASE, особенно в руководствах «Администрирование системы» и «Быстродействие и настройка».

3 Оценка сложности миграции

Для выполнения проекта миграции БД исключительно важно заранее получить точную оценку сложности этого процесса. Под «сложностью» здесь понимается способ сопоставления специфических функций Oracle с набором функций Sybase ASE.

Вначале следует тщательно изучить существующую систему Oracle и составить список всех типов используемых функций, характерных для Oracle, с указанием того, сколько раз они встречаются. Для каждой используемой функции следует определить, в какую категорию она попадает:

- **Простая конверсия**
 Функциональность или команду (оператор) системы Oracle можно сопоставить и непосредственно преобразовать в (почти) идентичную функцию Sybase ASE – для этого не потребуется менять синтаксическую структуру, можно ограничиться лишь простым локальным изменением синтаксиса.
 Примеры: большинство сопоставлений типов данных (Oracle VARCHAR2 Sybase VARCHAR); простые операторы SELECT.
- **Частичное переписывание**
 Функцию или оператора системы Oracle можно сопоставить с частично эквивалентной функцией Sybase ASE – для этого может потребоваться значительное изменение синтаксической структуры и, возможно, частичное переписывание алгоритмов. Пример: последовательности Oracle столбцы идентичности Sybase ASE.
- **Полное переписывание**
 Для функции или оператора Oracle отсутствует прямой эквивалент в Sybase ASE – потребуется переписывание или повторная разработка алгоритмов либо отдельных частей приложений.
 Пример: технология Oracle Flashback; триггеры уровня строки в Oracle.

Разбиение функций Oracle, используемых в мигрируемой системе, по категориям помогает выявить области, где могут возникнуть наибольшие трудности при переходе. До принятия решения о начале миграции необходимо иметь четкое представление о количестве вхождений функций из категорий «Частичное переписывание» и «Полное переписывание», а также о том, какие усилия потребуется приложить для переноса этих функций (особенно из категории «Полное переписывание»).

Чтобы помочь оценить сложность процесса, ниже приводятся три перечня контрольных вопросов, соответствующих данным категориям. В них представлен целый ряд функций Oracle. Следует иметь в виду, что в конкретной системе могут использоваться другие функции Oracle, которых нет в приведенных перечнях; их также следует учесть.

В данных перечнях приведено лишь краткое описание функций Oracle. В главе 11 содержатся расширенные варианты перечней: в них указаны соответствующие эквиваленты Sybase ASE для каждой функции Oracle.

3.1 Контрольный перечень Oracle: типы данных

Проверьте типы данных, которые используются в текущем приложении Oracle (см. раздел 4.7).

Соображения о миграции особых типов данных приведены в разделе 6.5.

3.2 Контрольный перечень Oracle: категория «Простая конверсия»

Кол-во найденных случаев	Контрольный перечень Oracle: категория «Простая конверсия»
	Подключение к схеме Oracle
	В конце некоторых изученных процедур в Oracle стоит знак «косая черта» (слеш)
	Точка с запятой (как разделитель операторов в PL/SQL)
	Таблица DUAL в Oracle
	Имя оператора точки сохранения SET SAVEPOINT

Переменная/Определения параметров; синтаксис имен
Задание значения по умолчанию в определении переменной
Несколько определений с единственным ключевым словом DECLARE
Определения без ключевого слова DECLARE в разделе определений для сохраненных процедур/функций
Назначение переменных
Перенос данных таблицы в переменную
Константы
%TYPE обозначает тип данных в столбце существующей таблицы
Динамический SQL (выполняемый немедленно)
Циклы LOOP/END LOOP
Циклы FOR
Циклы CURSOR FOR
Синтаксис внешнего присоединения Oracle
SET TRANSACTION READ WRITE
ALTER TABLE mytable TRUNCATE PARTITION partition_name
CREATE OR REPLACE PROCEDURE (или FUNCTION)
ALTER PROCEDURE (или FUNCTION)
CREATE PROCEDURE... IS...
Выполнение сохраненной процедуры с поименованными параметрами (параметр => значение)
Выполнение сохраненной процедуры с позиционными параметрами (:var)
Выполнение сохраненной процедуры
Определение функции SQL с ключевым словом DETERMINISTIC
Выполнение функции SQL
DECLARE CURSOR имя курсора IS...
Курсоры Oracle
Атрибут курсора %ISOPEN
Атрибуты курсоров %FOUND, %NOTFOUND

Атрибут курсора %ROWCOUNT

Триггеры AFTER (на уровне операторов)

SQL%ROWCOUNT

Тип данных BOOLEAN (только для переменных PL/SQL)

Оператор MERGE

Разделенные таблицы с композитным разделением

Типы данных native PL/SQL для оптимизации производительности (только для переменных PL/SQL)

BINARY_INTEGER

BINARY_DOUBLE

BINARY_FLOAT

IF-THEN-ELSE

Множественные операторы в операциях перехода IF-THEN-ELSE

Условное тестирование на основе подзапроса EXISTS

Оператор конкатенации строк: ||

userenv('sessionid')

MOD (X,Y)

CEIL ()

TRUNC (число)

SUBSTR ()

Функция SUBSTR () с двумя параметрами

LENGTH ()

CHR ()

TO_CHAR(выражение)

TO_CHAR (выражение, часть даты)

TO_CHAR (выражение, строка, задающая формат)

TO_NUMBER (выражение)

Функции и расчеты даты/времени

SYSDATE, SYSTIMESTAMP

TRUNC (дата/время [,единица])
LAST_DAY()
Функция NVL ()
Несогласованное использование символов верхнего или нижнего регистров для идентификаторов (регистр не имеет значения для идентификаторов в системах Oracle)
Идентификаторы, которые являются зарезервированными словами в Sybase ASE (см. раздел 4.8)
Функция INSTR () с двумя параметрами
Производные таблицы (называемые также inline views – «встроенные представления») без соотнесения имени
ALTER TABLE... SPLIT PARTITION. ALTER TABLE... MERGE PARTITIONS...
Подсказки оптимизатору Oracle (hints)

3.3 Контрольный перечень Oracle: категория «Частичное переписывание»

Для приведенных ниже функций Oracle возможна миграция с получением частично эквивалентных функций Sybase ASE, хотя это может потребовать значительного изменения синтаксической структуры и частичного переписывания алгоритмов.

Кол-во найденных случаев	Контрольный перечень Oracle: категория «Частичное переписывание»
	Указатели связей с базами данных
	Внешние таблицы
	Последовательности
	Функции SQL, задаваемые пользователем в виде табличных значений
	Синонимы
	Комментарии для объектов базы данных
	Индексы растровых отображений
	Временные таблицы
	Вложенные таблицы
	Таблицы объектов
	%ROWTYPE
	Определение типа записи PL/SQL путем нумерации полей с помощью IS RECORD OF или TYPE... IS RECORD

IS TABLE OF		
Нецелочисленное значение RETURN в сохраненной процедуре		
Задаваемые пользователем пакеты		
Переполнение в сохраненных процедурах		
Обработка исключений PL/SQL; определение средств обработки исключений		
SQLCODE, SQLERRM		
RAISE_APPLICATION_ERROR		
Кодирование столбцов		
Локаторы LOB		
Сжатие данных		
Вызовы пакетов DBMS_*		
Извлечение данных из сохраненных процедур в приложение-клиент с использованием пакета DBMS_OUTPUT		
Загрузчик SQL*Loader (sqlldr)		
Глобальные переменные (в пакете PL/SQL)		
Конструкция INTERSECT		
Конструкция MINUS		
Специфические операторы SQL		
AS OF	CROSS	IGNORE
AS OF TIMESTAMP	CUBE	ITERATE
CONNECT BY	FOR	NATURAL
DIMENSION	KEEP	NULLS
DIMENSION BY	MAIN	NULLS FIRST
EXCLUDE	MODEL	NULLS LAST
GROUPING SETS	NAV	ROLLUP
INCLUDE	NOCYCLE	SIBLINGS
MEASURES	NOWAIT	SINGLE
RETURN ALL ROWS	ON	REFERENCE
RETURN UPDATED	ONLY	LOCKED
ROWS	RULES	START WITH
PARTITION BY	SAMPLE	UNIQUE
REFERENCE	SEED	UNPIVOT
SYSTIMESTAMP	SKIP	WAIT
INITCAP (строчное выражение)		
Функция INSTR () с тремя или четырьмя параметрами		
Функция NVL2 ()		

Функция DECODE ()
Первичный ключ и внешний ключ с разными типами данных, разной точностью и (или) масштабом (для числовых типов данных) либо разной длины (для символьных типов данных)
Функции SQL, в которых последним оператором не является RETURN
Производные таблицы (называемые также inline views – «встроенные представления») с использованием синтаксиса «with»
UNION в курсорах
Директивы PRAGMA
ROWID
ROWNUM

3.4 Контрольный перечень Oracle: категория «Полное переписывание»

Для перечисленных ниже функций Oracle не существует прямых эквивалентов в Sybase ASE. Поэтому потребуется переписать или повторно разработать алгоритмы либо отдельные части приложений.

Кол-во найденных случаев	Контрольный перечень Oracle: категория «Полное переписывание»
	<p>Oracle MVCC (мультиверсионный контроль параллельного выполнения операций; «писатели не блокируют читателей, читатели не блокируют писателей»)</p> <p>Аспекты, представляющие интерес:</p> <ul style="list-style-type: none"> • Приложения или запросы, предполагающие отсутствие блокировки MVCC • Длительно выполняемые транзакции • DDL в транзакциях • SET TRANSACTION READ ONLY • SQL*Plus – автоматическая фиксация транзакции/фиксация при выходе
	SQL*Plus
	Триггеры BEFORE
	Триггеры уровня строки (BEFORE и AFTER)
	Несколько триггеров для типа DML в таблице
	Реализованные отображения
	REF CURSOR
	Запросы в многооконном режиме (SELECT... OVER (...))
	Параметры OUT/IN OUT для функции SQL

Недетерминированные функции SQL (функции, результат применения которых не зависит от параметров ввода)

Агрегатные функции SQL

Тип данных BFILE

Средства Oracle Streams; Oracle Data Guard

Oracle RAC для обеспечения высокой доступности

Технология Oracle Flashback

Управление планами Oracle SQL Plan Management

AWR (Automatic Workload Repository – автоматический репозиторий рабочей нагрузки)

Расширенные функции управления очередями Oracle Advanced Queuing

Пакеты PL/SQL для доступа к Интернету

OWA_CUSTOM, OWA_CX, OWA_OPT_LOCK, OWA_SEC, OWA_TEXT, OWA_UTIL

Формы Oracle

4 Миграция схемы базы данных

Первый шаг в миграции базы данных Oracle на сервер Sybase ASE – перенос схемы базы данных. Здесь понятие «схема» базы данных относится к физической модели данных. Другими словами, этот термин определяет структуру базы данных, в частности таблиц, столбцов, индексов, отображений, типов данных и т. д., выражаемую обычно в SQL DDL (Data Definition Language – язык описания данных), например в операторах типа «создать таблицу».

Отметим, что существуют терминологические разногласия в связи с использованием понятия «схема»:

- в качестве общего понятия, относящегося к базам данных, схема определяет структуру БД (как описано выше), независимо от того, кто из пользователей БД является владельцем объектов.
- В системах Oracle схема является ключевым понятием. Она представляет собой совокупность объектов базы данных (таблицы, отображения, сохраненные процедуры, триггеры и т. д.), которыми владеет определенный пользователь. В ходе миграции потребуются принять решение о том, как сопоставить схемы Oracle и ASE (см. раздел 4.5 для получения более подробных сведений).
- В Sybase ASE под схемой обычно понимают общую концепцию схемы базы данных.
- Примечание: для полноты картины отметим, что в ASE также существует команда «create schema authorization», которая создает таблицы, отображения и связанные с ними настройки доступа в качестве транзакционного блока. Однако эта команда используется редко, и мы не будем упоминать о ней далее в данном Руководстве.
Условимся, что в данном Руководстве термин «схема Oracle» будет обозначать схему в том смысле, который принят в системах Oracle. Во всех прочих случаях термин «схема» будет относиться к общей концепции схемы БД, как указано выше.

Замечания: ни один из описанных в данной главе методов не позволяет конвертировать код PL/SQL Oracle в Transact-SQL Sybase, что требуется для конвертирования сохраненных процедур, триггеров и функций SQL. Такие возможности предоставляет инструмент SwisSQL (дополнительные сведения приведены в других разделах данного Руководства).

4.1 Определение схемы Oracle

При миграции схемы БД из Oracle на ASE необходимо, прежде всего, получить схему Oracle, а затем конвертировать ее в формат и синтаксис, пригодные для использования в Sybase ASE.

Для получения схемы Oracle существуют два метода:

- Использование имеющихся скриптов DDL, из которых ранее была создана схема Oracle. Обычно в хорошо организованных средах эти скрипты хранятся в репозитории исходных кодов с контролем версий.
- Получение схемы Oracle обратным путем из реальной базы данных Oracle.

4.1.1 Использование существующих скриптов DDL

Если не прибегать к инструментам декомпиляции для миграции схемы, то на начальном этапе идеально будет использовать существующие скрипты DDL – в этом случае для получения схемы Oracle не потребуется никакой дополнительной работы. Однако возникает вопрос: можно ли гарантировать, что эти скрипты – актуальные и идентичные реальной базе данных Oracle? Нередко приходится сталкиваться с ситуациями, когда в схему базы данных вносятся изменения, а скрипты DDL в репозитории не обновляются. Ясно, что если полагаться на неправильные скрипты DDL, то появление проблем в процессе миграции неизбежно.

При наличии скриптов Oracle DDL следующим шагом будет конвертирование типов данных в Sybase ASE. В разделе 4.7 приводится сопоставление типов данных Oracle и Sybase ASE. Кроме того, некоторые аспекты схемы Oracle требуют особого внимания (см. раздел 4.4).

4.1.2 Декомпилирование существующей схемы

Альтернативный способ использования существующих скриптов – получение схемы Oracle путем декомпиляции реальной базы данных Oracle. В этом случае трудозатраты будут выше и потребуются специальные инструменты, однако сгенерированный DDL будет правильным – и это главное преимущество данного подхода.

Если существующие скрипты использовать нельзя или нет уверенности в их правильности, мы рекомендуем использовать для декомпилирования и миграции схемы БД инструмент Sybase PowerDesigner. Он позволяет декомпилировать таблицы, индексы и т. д., автоматически конвертировать типы данных Oracle в их эквиваленты в ASE, поэтому данный метод миграции схемы является самым быстрым и наиболее эффективным.

- В разделе 4.2 описано, как использовать PowerDesigner.
- В разделе 4.3 рассмотрен возможный подход к декомпиляции схемы без применения PowerDesigner.

4.2 Использование Sybase PowerDesigner для миграции схемы базы данных

Можно поспорить с тем, что Sybase PowerDesigner является наиболее совершенным из существующих на рынке инструментов моделирования данных. Это автономное средство, работающее в ОС Windows. PowerDesigner поддерживает более 30 типов баз данных, включая БД Oracle и Sybase ASE.

Для получения более подробной информации о PowerDesigner перейдите по ссылке <http://www.sybase.com/powerdesigner>.

С помощью PowerDesigner относительно несложно выполняется декомпиляция большинства схем Oracle и их конвертация в Sybase ASE. Центральное понятие, используемое в PowerDesigner, – модель физических данных PowerDesigner или PDM (Physical Data Model). Эта модель не зависит от базы данных и ее можно конвертировать в диалект SQL DDL каждой поддерживаемой базы данных.

4.2.1 Этапы конвертирования схемы с помощью PowerDesigner

В этом разделе показано, как использовать PowerDesigner для конвертирования схемы базы данных из Oracle в Sybase ASE. Прежде всего, следует установить следующие продукты:

- Sybase PowerDesigner – последний выпуск, не ниже версии 15.0;
- драйвер Oracle ODBC.

Последовательность шагов декомпилирования схемы (для таблиц, индексов, отображений, ограничений):

1. Запустите PowerDesigner.
2. Выберите **File>Reverse Engineer>Database**, чтобы открыть диалоговое окно новой модели физических данных.
3. Выберите в списке базу данных Oracle и нажмите кнопку «ОК».
4. В открывшемся диалоговом окне Database Reverse Engineering (Декомпиляция базы данных) выберите кнопку-переключатель Using an ODBC data source (Использование источника данных ODBC).
5. Выберите инструмент **Connect to an ODBC Data Source** (Подключиться к источнику данных ODBC). После этого откроется диалоговое окно Connect to an ODBC Data Source (Подключение к источнику данных ODBC).
6. Выберите кнопку-переключатель Machine data source (Источник данных машины), затем укажите в списке источник данных Oracle.
7. Введите идентификатор пользователя (ID) и пароль, затем нажмите кнопку Connect (Подключиться), чтобы вернуться в диалоговое окно Database Reverse Engineering (Декомпиляция базы данных).
8. Нажмите кнопку «ОК». Откроется диалоговое окно ODBC Reverse Engineering (Декомпиляция ODBC). Выберите в этом окне объекты для декомпиляции. По умолчанию указываются только таблицы и триггеры.
9. Нажмите кнопку «ОК», чтобы начать процесс декомпиляции. После его завершения появится подтверждающее сообщение.
10. Выберите установленный драйвер ODBC и выполните подключение.
11. Выберите все элементы и создайте физическую модель данных (PDM).
12. Сохраните PDM, выбрав **File > Save As** (в качестве имени файла используйте имя базы данных в Sybase ASE).

Теперь, когда схема извлечена из Oracle в PDM, измените конечную DBMS на Sybase ASE:

1. Выберите **Database > Change Current DBMS**
2. В качестве конечной DBMS выберите в раскрывающемся списке Sybase AS Enterprise 15.0.

Теперь сгенерируйте язык DDL, совместимый с Sybase ASE, из PowerDesigner PDM; при этом типы данных автоматически конвертируются в типы данных для Sybase ASE:

1. Выберите **Database > Generate Database**. Откроется диалоговое окно Database Generation (Создание базы данных).
2. Введите название конечного каталога и имя файла скрипта в поля Directory (Каталог) и File Name (Имя файла).

3. Выберите кнопку-переключатель Script generation (Создание скрипта).
4. На экране отобразится процесс создания скрипта и будет указан синтаксис для запуска скрипта. По окончании процесса откроется окно Result (Результат). В нем будет показан путь к файлу созданного скрипта. Чтобы открыть скрипт в текстовом редакторе, нажмите кнопку Edit (Правка). Чтобы закрыть окно Result, нажмите кнопку Close (Закреть).
5. Просмотрите все предупреждения и сообщения об ошибках; при необходимости внесите исправления и создайте скрипт заново.

Некоторые аспекты миграции схем нельзя обработать в PowerDesigner, эти операции придется выполнять вручную. Соответствующая процедура описана в разделе 4.4.

В завершение запустите созданный скрипт DDL в Sybase ASE и проверьте его на наличие ошибок.

4.3 Декомпилирование схемы Oracle без использования PowerDesigner

Декомпилирование схемы без использования Sybase PowerDesigner можно выполнить несколькими способами:

- С помощью команды Oracle DESC для всех объектов базы данных с последующим преобразованием полученных результатов в DDL. Для этого потребуются написание значительного объема кода вручную.
- С помощью TOAD – недорогого инструмента, широко применяемого во многих средах Oracle. Он позволит извлечь определения объектов, а затем вручную конвертировать типы данных Oracle в соответствующие эквиваленты ASE. Этот процесс будет довольно трудоемким при наличии большого количества таблиц.

После декомпиляции схемы Oracle необходимо конвертировать Oracle DDL в синтаксис Sybase ASE, включая преобразование типов данных Oracle в типы данных Sybase ASE. В разделе 4.7 приводится сопоставление типов данных Oracle и Sybase ASE.

Кроме того, некоторые аспекты схемы Oracle требуют особого внимания (см. раздел 4.4).

4.4 Особые случаи миграции схем

Следующие аспекты схем требуют особого внимания:

- В системах Oracle количество столбцов в таблице может быть больше, чем в Sybase ASE (предел зависит от размера страницы сервера ASE и от схемы блокировки таблицы). Если предельное значение будет превышено в Sybase ASE, то при создании таблицы возникнет ошибка. В этом случае необходимо либо увеличить размер страницы сервера ASE, либо разбить таблицу по вертикали на несколько таблиц. При этом придется соответствующим образом изменить все запросы, обращенные к таблице.
- Если длина столбца превышает максимально допустимую в Sybase ASE (она определяется размером страницы на сервере ASE и схемой блокировки таблицы), то придется разбить такие столбцы и поместить их в дополнительные таблицы. Аналогичным образом придется изменить все запросы, обращенные к таблице.
- PowerDesigner конвертирует тип данных Oracle BFILE в тип данных image в Sybase ASE. Поскольку BFILE – тип данных для хранения указателей (ссылок) на внешний двоичный файл, который хранится вне базы данных, то в этом случае функциональная эквивалентность отсутствует и может потребоваться изменить приложение. Если нужен другой тип данных ASE, например содержащий имя файла из внешнего хранилища, то изменение надо будет сделать вручную.
- PowerDesigner 15.x не может автоматически конвертировать тип данных timestamp Oracle в тип данных bigdatetime в ASE – данную операцию нужно выполнить вручную. PowerDesigner 16.0 (выпуск ожидается в августе 2011 г.) не имеет этого ограничения и конвертирование будет выполнено в автоматическом режиме.
- PowerDesigner 15.x не может декомпилировать пользователей Oracle или сведения о безопасности (права доступа). PowerDesigner 16.0 (выпуск ожидается в августе 2011 г.) не имеет указанных ограничений и может обрабатывать данные аспекты.
- Поскольку зарезервированные слова SQL в Oracle и Sybase ASE отличаются, то перед миграцией схемы базы данных необходимо проверить все объекты Oracle на предмет отсутствия зарезервированных слов Sybase ASE. Следует в первую очередь заменить все идентификаторы Oracle, которые совпадают с зарезервированными словами Sybase ASE. Полный список зарезервированных слов в Sybase ASE можно найти в «Adaptive Server Enterprise->Reference Manual: Building Blocks (Справочное руководство: компоненты)->Reserved Words

(Зарезервированные слова)». Также см. раздел 4.8 – в нем описаны запросы, используемые для поиска вхождений ключевых слов в базе данных Oracle.

- Сопоставление задаваемых пользователем типов данных Oracle с типами данных ASE может оказаться непростым делом, требующим выполнения значительного объема ручных работ. Ключевым фактором успешной миграции типов данных, задаваемых пользователем, является полное понимание типов данных, на которых создана база. Следует помнить, что задаваемые пользователем типы данных могут быть вложенными. Для Oracle задание пользовательских типов данных – это дополнительная опция БД, которая не нашла широкого применения.

4.5 Сопоставление схемы Oracle с базами данных Sybase ASE

В Sybase ASE нет идентичной интерпретации понятия «схема», как в случае со схемой Oracle. При миграции схемы Oracle в Sybase ASE можно использовать два основных метода для сопоставления этих схем.

В качестве примера предположим, что у нас есть два пользователя Oracle – john и bill, которые являются владельцами схемы Oracle, и в каждой схеме имеется таблица с названием salesdetails (сведения о продажах).

Возможны следующие варианты действий:

- Самый прямой способ миграции – соотнесение каждой схемы Oracle с отдельной базой данных ASE, причем владельцем каждой базы данных (dbo) будет соответствующий пользователь. Это приведет к тому, что появятся две базы данных ASE с названиями john_db и bill_db (разумеется, можно выбрать другие имена), владельцами которых будут пользователи с логинами ASE для входа в систему john и bill, соответственно; каждая база данных будет иметь таблицу с названием salesdetails, владельцем которой является пользователь dbo-базы данных (полное название таблицы – dbo.salesdetails).

Однако это приведет к тому, что количество баз данных в ASE будет равняться количеству пользователей-владельцев схемы Oracle, а последних может быть очень много. Несмотря на то, что на сервере ASE может размещаться до 32 786 баз данных, в высшей степени нецелесообразно, с точки зрения администрирования базы данных в будущем, иметь более 20-50 БД.

- Соотнесение всех схем Oracle с единственной базой данных ASE с помощью модели мультипринадлежности. Это означает, что для идентификации владельца каждого объекта в базе данных применяется пользователь базы данных ASE (он имеет логин для входа на сервер ASE, что эквивалентно пользователю Oracle). Такой подход приведет к созданию более легкой в управлении системы ASE, так как количество баз данных ASE уменьшится.

В этом случае появится единственная база данных ASE, допустим, sales_db, в которую добавлены логины ASE john и bill в качестве пользователей базы данных. Каждый пользователь сможет создать таблицу salesdetails (полное название –john.salesdetails и bill.salesdetails).

Можно воспользоваться любым вариантом; с технической точки зрения в ASE они оба равнозначны. Однако модель мультипринадлежности лучше подходит для применяемых в ASE способов резервного копирования и восстановления данных.

Следует отметить, что модели мультипринадлежности иногда ошибочно считаются уязвимыми с точки зрения безопасности: ведь пользователю с логином bill легче получить доступ к таблицам пользователя john, поскольку они находятся в одной базе данных ASE. Однако это неоправданное опасение: если руководствоваться стандартными передовыми практиками в области защиты данных в ASE, то можно полностью гарантировать безопасность.

Отметим, что базы с мультипринадлежностью имеют одну особенность: резервная копия базы данных содержит данные всех пользователей. Если это нежелательно, например, каждый пользователь хочет иметь резервную копию своей собственной базы данных, то следует воспользоваться первым вариантом (отдельные базы данных ASE для каждого пользователя).

Наконец, возможен случай, когда существует только одна схема Oracle. Здесь нет необходимости классифицировать таблицы ASE с помощью имени владельца, поскольку владельцем всех таблиц будет dbo-пользователь.

4.6 Терминология Oracle–Sybase, связанная со схемами

Ниже приведено сопоставление терминологии верхнего уровня для Oracle и Sybase. Эта таблица не предназначена для использования в процессе миграции, ее следует рассматривать только в качестве руководства по терминологии верхнего уровня.

Oracle	Sybase ASE
База данных	Сервер базы данных
Схема	База данных и объекты, которыми владеет один и тот же пользователь.
Табличное пространство	База данных
Пользователь	Пользователь
Роль	Роль
Таблица	Таблица
Временная таблица	Временная таблица
Отображение	Отображение
Реализованное отображение	Прямой эквивалент отсутствует
Кластер	Кластеризованный индекс для таблиц блокировки страниц данных
Индекс	Неуникальный индекс
Таблица, организованная с помощью индекса	Таблица с кластеризованным индексом
Ограничение проверки на уровне столбцов	Ограничение проверки на уровне столбцов
Столбец по умолчанию	Столбец по умолчанию
Уникальный ключ	Уникальный ключ или свойство идентичности для столбца
Первичный ключ	Первичный ключ
Внешний ключ	Внешний ключ
Ограничения	Ограничения
Процедура PL/SQL	Сохраненная процедура T-SQL
Функция PL/SQL	SQL-функция, задаваемая пользователем, T-SQL (SQL UDF)
Триггеры	Триггеры
Пакет	Прямой эквивалент отсутствует
Последовательности	Им частично соответствует свойство идентичности для столбца или таблица значений специальных ключей

Моментальный снимок	Прямой эквивалент отсутствует
Связи баз данных, внешние таблицы	Прокси-таблицы и удаленные серверы
Процедура	Сохраненная процедура
Синоним	Аналогичная функциональность с отображениями для синонимов таблиц и отображений. Все другие ссылки на синонимы следует заменить на полностью классифицированные объектные строки.

4.7 Сопоставление типов данных Oracle и Sybase ASE

В следующей таблице соотнесены типы данных Oracle и Sybase ASE. В большинстве случаев имеется прямое соответствие типов данных.

Для таких типов данных Oracle, как CHAR, VARCHAR2 и RAW, возможность соотнесения определяется размером страницы сервера ASE. С технической точки зрения это объясняется следующим: в ASE необходимо, чтобы строка и, следовательно, каждый столбец помещались на странице БД ASE. По умолчанию в ASE используется страница сервера размером 2 Кб (допускается также использование страниц размером 4 Кб, 8 Кб и 16 Кб). Максимально допустимая длина столбца на странице сервера ASE зависит от различных факторов, например от его длины (фиксированная или переменная), а также от схемы блокировки таблицы ASE. Для просмотра полных сведений введите в ASE команду dbcc serverlimits.

Oracle	Описание	Sybase ASE	Комментарии/Когда использовать
NUMBER(x)	Типы данных Oracle NUMBER(x) с 0 десятичных разрядов можно конвертировать в эквивалентные типы данных Sybase ASE	BIGINT	длина типа данных NUMBER > 10
		INTEGER	длина типа данных NUMBER – от 6 до 10, значение данных <= 2 млрд
		SMALLINT	длина типа данных NUMBER – от 4 до 5, значение данных <= 32 767
		TINYINT	длина типа данных NUMBER – от 2 до 3, значение данных <= 255
NUMBER(x,y)	эти типы данных Sybase ASE можно использовать вместо указанных выше соответствий	NUMERIC(x,y) DECIMAL(x,y)	точная, один в один, трансляция типа данных Oracle NUMBER
		MONEY SMALLMONEY	типы данных MONEY и SMALLMONEY служат для хранения данных в денежном формате; точность – 4 разряда справа от десятичной запятой и 16/6 разрядов слева от нее для MONEY/SMALLMONEY, соответственно

FLOAT	максимальная точность FLOAT в Oracle около 38 цифр	DOUBLE	точность реальных значений > 15
		FLOAT	точность реальных значений <= 15
CHAR(x)	максимальный размер CHAR в Oracle составляет 2000 байт	CHAR(x)	если размер страницы ASE составляет 4 Кб или больше, размер страницы ASE – 2 Кб, x <= 1958
		TEXT	если не выполняется ни одно из указанных условий
VARCHAR2(x)	максимальный размер VARCHAR2 в Oracle составляет 4000 байт (для столбцов)	VARCHAR(x)	если размер страницы ASE составляет 8 Кб или больше; если размер страницы ASE составляет 4 Кб и x <= 3988; если размер страницы ASE составляет 2 Кб и x <= 1948
		TEXT	если не выполняется ни одно из указанных условий
DATE	точность даты/времени в Oracle в пределах одной секунды	DATETIME	точность типа данных DATETIME в Sybase ASE составляет 1/300 часть секунды
TIMESTAMP [WITH [LOCAL] TIME ZONE]	точность типа данных TIMESTAMP в Oracle составляет 1/100000000 долю секунды	BIGDATETIME	точность BIGDATETIME в Sybase ASE составляет 1 микросекунду. ASE не поддерживает временные зоны
ROWID	псевдо-столбец в Oracle, не является действительным типом данных	NUMERIC IDENTITY	См. также ROWID на стр. 60
CLOB	максимальный объем для хранения CLOB в Oracle составляет 128 Тб	TEXT	В Sybase ASE на каждый столбец приходится не более 2 Гб; в IQ – до 2 Пб
NCLOB	максимальный объем для хранения NCLOB в Oracle составляет 128 Тб	UNITEXT	В Sybase ASE на каждый столбец приходится не более 2 Гб; в IQ – до 2 Пб
BLOB	максимальный объем для хранения BLOB в Oracle составляет 128 Тб	IMAGE	В Sybase ASE на каждый столбец приходится не более 2 Гб; в IQ – до 2 Пб
LONG	максимальный объем для хранения LONG в Oracle составляет 2 Гб	TEXT	
RAW(x)	максимальная точность типа данных RAW в Oracle составляет 2000 байт	BINARY(x)	если размер страницы ASE составляет 4 Кб или больше; если размер страницы ASE – 2 Кб и x <= 1954
		VARBINARY(x)	
		IMAGE	если не выполняется ни одно из указанных условий

LONG RAW	максимальный объем для хранения LONG RAW в Oracle составляет 2 Гб	IMAGE	
CHAR(1)	если это упакованный битовый столбец с поддержкой функций PL/SQL set/unset/retrieve/query	BIT	
BFILE	BFILE хранит указатель (ссылку) на двоичный файл за пределами базы данных		Прямой эквивалент отсутствует

4.8 Поиск зарезервированных слов в Sybase ASE и ключевых слов в Oracle

До выполнения миграции схемы Oracle или сохраненной процедуры Oracle необходимо провести проверку на наличие зарезервированных (ключевых) слов, которые могут вызвать проблемы или которые нельзя перенести.

С помощью приведенного фрагмента кода можно выполнить сканирование любого объекта в базе данных Oracle на наличие ключевых слов. Сканирование возвращает имя и владельца объекта, а также тип объекта:

```
SELECT owner, name, type
FROM dba_source WHERE instr(UPPER(text), UPPER('<keyword>')) > 0
```

Иногда бывает важно получить точный код и номер строки вхождения в сохраненной процедуре, функции или триггере. Отметим, что следующий запрос может вернуть большое количество данных:

```
SELECT owner, name, type,line, text
FROM dba_source WHERE instr(UPPER(text), UPPER('<keyword>')) > 0
```

Приведенные запросы следует выполнить для всех зарезервированных и ключевых слов Sybase ASE. Полный список зарезервированных слов в Sybase ASE можно найти в «Adaptive Server Enterprise->Reference Manual: Building Blocks (Справочное руководство: компоненты)->Reserved Words (Зарезервированные слова)».

4.9 Выбор схемы блокировки для таблиц в Sybase ASE

Для каждой таблицы в базе данных ASE можно выбрать одну из трех схем блокировки: allpages, datapages или datarows.

Схема allpages представляет собой старейшую схему блокировки, которая поставляется в ASE по умолчанию. Она более эффективна для некоторых типов операций. Схемы блокировки страниц данных и особенно строк данных обеспечивают более эффективные характеристики параллельного выполнения операций. Преимущества параллельного выполнения операций нелишне будет упомянуть при миграции из Oracle в Sybase ASE из-за разницы в процедуре обработки транзакций (см. главу 8).

По умолчанию рекомендуется выбирать схему блокировки datapages или datarows в Sybase ASE. Схема datapages более эффективна, однако datarows – лучше при параллельном выполнении операций (блокировка datarows называется также блокировкой на уровне строки). Переключение между datarows и datapages для существующей таблицы происходит мгновенно. Напротив, если потребуется изменить схему блокировки больших таблиц callpages на datarows или datapages, то это увеличит время простоя, что связано с необходимостью полной конвертации таблицы и всех ее индексов.

4.10 Таблица DUAL в Oracle

В Oracle для таблиц всегда следует выполнять оператор SELECT, даже при получении системной информации, например, даты/времени. С этой целью в Oracle была создана таблица DUAL. Получение системной даты посредством SQL имеет следующий вид в Oracle:

```
SELECT sysdate FROM DUAL
```

Sybase ASE поддерживает операторы SELECT, не имеющие условия FROM. Тот же самый запрос в Sybase ASE выглядит так:

```
SELECT getdate()
```

Чтобы избежать переписывания существующих операторов SELECT, использующих таблицу DUAL, можно создать таблицу с названием DUAL в ASE, которая всегда содержит только одну строку:

```
create table DUAL (dummy_col char(1) unique check (dummy_col='X'))
insert DUAL values ('X')
go
```

Если таблица Sybase ASE чувствительна к регистру символов (см. раздел 5.2), может потребоваться создание дополнительных таблиц с названиями dual, Dual и т. д., в зависимости от того, как варьировалось разработчиками Oracle написание названия DUAL. Можно исправить исходный код Oracle PL/SQL так, чтобы он содержал только «DUAL», или полностью убрать все ссылки на DUAL.

5 Миграция аспектов серверного уровня

Архитектура сервера баз данных и способ его настройки и управления в Oracle и Sybase ASE существенно отличаются. В этой главе перечислены некоторые аспекты миграции, которым следует уделить внимание, однако данный перечень не полон. Читатель должен обязательно сверяться с документацией Sybase, особенно с «Руководством по администрированию системы», для получения подробных сведений.

5.1 Набор символов

При создании нового сервера Sybase ASE необходимо выбрать набор символов, который будет использоваться этим сервером. Рекомендуется применять для ASE тот же набор символов, что и в базе данных Oracle.

Впоследствии можно менять набор символов для ASE, однако настоятельно рекомендуем этого избегать и сразу же выбрать нужный набор, до начала миграции любых аспектов Oracle в ASE.

5.2 Чувствительность к регистру сервера баз данных (порядок сортировки)

Разница между Oracle и Sybase ASE в том, что Oracle не различает символы верхнего и нижнего регистра, а Sybase ASE – различает по умолчанию. Чувствительность к регистру ASE относится к идентификаторам и сравнениям данных; на ключевые слова SQL это не распространяется.

ASE можно сделать нечувствительным к регистру, если установить порядок сортировки, не зависящий от регистра.

Необходимо принять решение о том, должен ли сервер ASE различать состояние регистра. С технической точки зрения в ASE оба варианта равноценны.

На практике решение может диктоваться тем, что код SQL в приложении Oracle может содержать символы верхнего и нижнего регистра. Например, код Oracle SQL по-разному ссылается на таблицу TEST в Oracle – все приведенные случаи относятся к одной и той же таблице:

```
select * from TEST
select * from Test
select * from test
```

Если использовать в Sybase ASE порядок сортировки, не зависящий от состояния регистра, то приведенные операторы SQL изменять не нужно. Если же использовать установленный в ASE по умолчанию порядок сортировки с чувствительностью к регистру, то во всех ссылках на таблицу необходимо использовать одинаковую последовательность заглавных и строчных букв. В противном случае будут сгенерированы сообщения об ошибке «Таблица не найдена».

5.3 Параметры конфигурации сервера

В Oracle параметры конфигурации сервера и базы данных хранятся в файле инициализации (init.ora) или в файле параметров сервера (spfile). Эти параметры охватывают различные наборы ресурсов, включая память, процессы, сеть, диск, устройство ввода/вывода, подключения, файлы, наборы символов и прочее.

Маловероятно, что конфигурационные параметры Oracle можно будет напрямую сопоставить с таковыми в Sybase ASE. Однако полезно знать о специфических настройках конфигурации Oracle, поскольку в некоторых случаях может понадобиться своего рода эквивалент в Sybase ASE.

С помощью одного из указанных ниже вариантов действий можно получить значения параметров Oracle, отличные от установленных по умолчанию:

- Конвертируйте файл параметров сервера (spfile) в файл параметров инициализации следующим образом:
CREATE pfile FROM spfile
- Создайте запрос к базе данных, выполнив следующий оператор:
SELECT name, value FROM sys.v\$spparameter
WHERE isspecified = 'TRUE'

5.4 Дисковая подсистема

Во многих инсталляциях Oracle используется помощник Automated Storage Manager (ASM) – Диспетчер автоматического сохранения. В Sybase ASE эквивалент ASM отсутствует. Управление хранением осуществляется с помощью команд T-SQL, Sybase Control Center (Центр управления Sybase) или с помощью Sybase Central – инструмента администратора базы данных Sybase с графическим интерфейсом.

Для Sybase ASE можно дать следующие инструкции верхнего уровня применительно к организации дисковой подсистемы:

- Для пользовательских баз данных используйте «сырые» устройства или устройства в виде файлов ОС с опцией `direction=true`. Никогда не используйте устройства файловой системы с `dsync=false` для пользовательских баз данных; устройства файловой системы с `dsync=true` применять можно, но это может значительно снизить производительность.
- Для временных баз данных обычно рекомендуется использовать устройства файловой системы с `dsync=false`.
- Для соответствующего уровня хранения рекомендуется RAID 0+1 или RAID 1+0. Избегайте использовать RAID 5 там, где выполняется много записей, например, в журнале транзакций ASE, за исключением случая, когда решение для хранения предусматривает применение долговременной кэш-памяти с целью буферизации записей.
- Для достижения максимальной ширины полосы пропускания дискового устройства ввода/вывода интенсивно считываемые и записываемые данные следует распределить по максимально возможному количеству физических дисков.

При создании конфигурации хранения следует учесть также множество дополнительных нюансов. Более подробную информацию можно найти в «Руководстве по администрированию системы» для Sybase ASE.

5.5 Миграция логинов пользователей

Существуют терминологические различия в трактовке понятия «пользователь» в Oracle и Sybase ASE.

- В Oracle на уровне экземпляров: понятие «пользователь» используется для авторизации и, кроме того, пользователь может быть владельцем схемы (а потому владеть объектами базы данных и иметь доступ к объектам БД).
- В Sybase ASE на уровне сервера: для авторизации используется логин, но он не связан с владением какими-либо объектами и не обеспечивает доступ к объектам.
В ASE существует особый логин `sa` (`super-user`), обозначающий супер-пользователя (аналогично учетной записи `SYS` в Oracle). Такой пользователь имеет доступ ко всем объектам базы данных, и этот доступ необходимо ограничить администрированием этой базы данных. В целях безопасности приложений никогда не следует использовать логин `sa` для подключения к серверу ASE.
- В Sybase ASE на уровне базы данных: пользователь, имеющий логин, может владеть объектами базы данных и иметь доступ к объектам базы данных.

Наиболее вероятный сценарий миграции из Oracle на Sybase ASE – миграция всех пользователей приложения в Oracle с присвоением им аналогичных логинов в Sybase ASE. Затем для каждого логина ASE создается соответствующий пользователь базы данных (обычно его имя совпадает с логином), чтобы этот логин получил доступ к пользовательской базе данных в ASE. Логин может иметь доступ к нескольким базам данных, если в каждой БД ASE создать соответствующего пользователя БД.

Другим вариантом является создание в каждой пользовательской БД ASE пользователя базы данных с именем `guest`. Однако предварительно следует тщательно оценить последствия такого решения с точки зрения безопасности.

Результирующая структура логинов и пользователей базы данных в ASE зависит от того, какие решения были приняты относительно миграции схемы Oracle в ASE (см. раздел 4.5).

5.5.1 Пароли пользователей

У каждого пользователя Oracle есть пароль. В ASE пароль есть у логина. Если известны пароли пользователей в Oracle, то в ASE можно задать такие же пароли; для логинов ASE надо будет задать новые пароли. Можно задать пустые пароли для логинов ASE.

5.6 Права доступа

Для декомпиляции прав доступа к объектам в базе данных Oracle мы рекомендуем использовать PowerDesigner 16. Если это невозможно, то права доступа придется конвертировать в эквивалентные права Sybase ASE вручную.

6 Миграция данных

В этом разделе описываются способы миграции данных из Oracle в Sybase ASE. Предполагается, что миграция схемы уже выполнена.

Главным осложняющим фактором является то, что в Oracle не предусмотрено никаких инструментов для выгрузки таблицы в неструктурированный файл такого формата, который можно прочитать с помощью средств сторонних разработчиков.

Существует несколько способов выполнения миграции данных. Поэтому при выборе подхода следует учитывать различные факторы:

- сложность выбранного решения;
- объем данных, которые требуется перенести;
- выделенное время для простоя системы, в течение которого требуется провести миграцию данных;
- необходимость ознакомления с новым программным обеспечением или инструментами для миграции данных;
- расходы на дополнительные лицензии на программное обеспечение.

По существу, для миграции данных существуют следующие возможности:

- **Выгрузка данных Oracle в неструктурированные файлы в формате ASCII, а затем загрузка этих файлов в ASE с помощью утилиты bcp.**

Если данные Oracle можно экспортировать в неструктурированный файл формата ASCII, то высокоскоростное средство загрузки ASE bcp сможет загрузить их в ASE. Поскольку Oracle не предусматривает никаких возможностей сделать это, пользователь должен либо использовать инструменты сторонних производителей, либо создавать собственную утилиту PL/SQL для извлечения данных из БД и записи их в неструктурированный файл.

На заметку: такую возможность часто рассматривают как привлекательную благодаря прозрачности процесса миграции. Все шаги четко просматриваются, их можно разрабатывать и тестировать в индивидуальном порядке. Разработка собственного инструмента PL/SQL для выгрузки данных Oracle – дело несложное, но работать он будет медленно. Поэтому данный вариант годится лишь для относительно небольших объемов данных. Использование инструментов стороннего разработчика увеличивает расходы из-за необходимости покупки лицензии.

- **Использование функциональности Enterprise Connect Data Access (ECDA), разработанной компанией Sybase для Oracle.**

ECDA – это продукт компании Sybase, который позволяет подключиться из базы данных ASE напрямую к базе данных Oracle и передать данные Oracle непосредственно в ASE. ECDA встраивается в механизм «прокси-таблиц» ASE.

На заметку: этот вариант используется в случае, если весь объем данных можно передать в предусмотренный для миграции интервал времени. Данный способ не подходит для передачи больших объемов данных. Его преимущество в том, что ECDA обеспечивает соответствие между типами данных Oracle и ASE, и что миграция полностью выполняется посредством SQL.

Для использования этого варианта необходимо приобрести продукт ECDA компании Sybase.

- **Использование Sybase Replication Server Heterogeneous Edition (RSHE) для Oracle**

Сервер репликации Sybase захватывает транзакции базы данных Oracle и переносит их в ASE, тем самым поддерживая БД ASE в актуальном состоянии. Кроме того, сервер репликации может скопировать полное содержимое таблиц Oracle в ASE, чтобы инициализировать репликацию данных («материализация системы репликации»).

На заметку: это единственное решение для миграции данных, которое позволяет БД Oracle продолжать работу во время выполнения перехода. В данном случае время простоя, когда выполняется миграция и приложения становятся недоступными (ведь их придется переключить с БД Oracle на БД ASE), не зависит от объема переносимых данных и может быть очень коротким (минуты, а не часы). Для использования данного варианта необходимо приобрести продукт Sybase Replication Server и научиться работать с ним.

Копирование исходных данных из Oracle в ASE с помощью сервера репликации не подходит для больших объемов данных. В этом случае исходную материализацию системы репликации можно выполнить с помощью одного из дополнительных вариантов, представленных ниже.

- **Использование средства ETL от стороннего разработчика, которое поддерживает Oracle и Sybase ASE.**

На заметку: этот вариант наиболее привлекателен, если инструмент ETL уже используется. В этом случае не придется приобретать дополнительное ПО только ради выполнения миграции.

Данный способ эффективен в случае, если весь объем данных можно передать в предусмотренный для миграции интервал времени. Он подходит и для очень больших объемов данных, хотя в этом случае время миграции значительно увеличится.

6.1 Выгрузка данных Oracle в ASCII-файлы; загрузка в ASE с помощью утилиты bcp

Утилита bcp для высокоскоростной загрузки данных позволяет загружать файлы практически любых типов в ASCII-формате в ASE. Однако, поскольку в Oracle не предусмотрено никаких средств экспорта данных в ASCII-файл, пользователь должен либо использовать инструменты сторонних производителей, либо разработать свою собственную утилиту PL/SQL для выгрузки данных из БД в неструктурированный файл. Пример такого инструмента – FACT.

6.1.1 Загрузка в ASE с помощью bcp

Приведем пример загрузки данных из ASCII-файла в таблицу ASE (с названием mydb.mytable) с помощью bcp:

```
bcp mydb.mytable in mytable.txt -Ulogin -Ppassword -Sserver -c
```

На практике в bcp нужно указать также, какие ограничители строки и столбца используются (параметры -r и -t), поскольку значения по умолчанию (CR и tab) могут встречаться в файле реальных данных (который в конечном итоге преобразуется в ASCII). При выгрузке данных в неструктурированные ASCII-файлы необходимо выбрать соответствующие ограничители.

Эффективнее всего bcp работает в том случае, если предварительно отбросить все индексы у загружаемых таблиц. В зависимости от размера и количества индексов, а также ширины таблицы последующее восстановление индексов может занять много времени для больших таблиц, поэтому данный вариант не всегда можно реализовать на практике.

Как правило, вместе с bcp лучше использовать для передачи по сети пакет большого размера (параметр -A; потребуется дополнительная настройка размера пакета на сервере ASE).

Для крупных таблиц можно использовать параметр -b, чтобы разбить нагрузку на несколько транзакций базы данных. Обычно такой подход сочетается с включением опции БД «trunclog on checkpt» (отбрасывать часть информации в журнале при достижении контрольного значения), чтобы журнал транзакций не переполнялся.

Чтобы загрузить только часть файла данных или загрузить столбцы в другой последовательности (иначе, чем в исходном файле), используется файл bcp-формата. Более подробные сведения о форматах файлов, а также о bcp можно найти в «Руководстве по использованию сервисных программ», которое входит в комплект документации ASE (<http://tinyurl.com/6883kx4>).

Настоятельно рекомендуем выполнять одновременно несколько bcp-операций (по одной для каждой загружаемой таблицы). Оптимальное количество параллельных bcp-операций определяется возможностями аппаратного обеспечения. Если имеется лишь одна крупная таблица (или небольшое число таблиц) для загрузки, то в этом случае можно запустить несколько bcp-операций и добавить разбиение таблицы по круговой схеме, с указанием начальной и конечной строк каждого конкретного сегмента.

Обратите внимание на то, что в Unix/Linux bcp может считывать данные из именованного канала (named pipe), созданного с помощью команды mkfifo. Если утилита, извлекающая данные в файл, может записывать их в именованный канал, то удастся сэкономить массу времени, поступив следующим образом:

1. Создайте именованный канал с помощью команды mkfifo в Unix/Linux.
2. Извлеките данные из системы Oracle, записывая их в именованный канал.
3. Не дожидаясь окончания извлечения данных, запустите bcp чтобы начать загрузку данных из этого именованного канала. Программа bcp будет считывать данные из именованного канала сразу же после их доставки утилитой извлечения данных и тут же вставлять их в ASE.

Время на передачу данных сократится благодаря тому, что процессы извлечения данных и их последующей загрузки выполняются параллельно, и займет столько времени, сколько потребуется для окончания более длительного процесса (время извлечения данных, время загрузки данных). Именно это позволит получить значительную экономию времени.

С более подробной информацией о bcp и именованных каналах можно ознакомиться по адресу <http://tinyurl.com/5urcfrt>.

6.1.2 Выгрузка из Oracle: FACT (инструмент стороннего производителя)

FACT (Fast Extract – быстрое извлечение) – это инструмент стороннего производителя, предназначенный для высокоскоростного экспорта данных Oracle в неструктурированный ASCII-файл и поддерживающий параллельный режим. Эти файлы можно использовать в качестве входных для утилиты Sybase ASE bcp.

Для получения более подробной информации об инструменте FACT перейдите по ссылке <http://www.iri.com/products/FACT>.

6.1.3 Выгрузка из Oracle: разверните собственную утилиту PL/SQL для экспорта данных Oracle

Если вы хотите выгрузить данные из таблиц Oracle в неструктурированные ASCII-файлы, используя только возможности Oracle, то вам придется создать собственную утилиту PL/SQL. Эта утилита будет извлекать данные из БД и передавать их в плоский файл. Для этого в PL/SQL используется команда DBMS_OUTPUT.put_line. Приведем пример экспорта двух столбцов таблицы emp используя «~» в качестве разделителя столбцов, а CR – в качестве разделителя строк. Результат выполнения представленного кода PL/SQL передается в неструктурированный файл:

```
DECLARE CURSOR emp_cur IS SELECT ename, sal FROM emp;
BEGIN
FOR emp_rec IN emp_cur
LOOP
DBMS_OUTPUT.PUT_LINE (emp_rec.ename || '~' || TO_CHAR (emp_rec.sal) );
END LOOP;
END;
/
```

Недостаток этого способа – невысокая скорость работы; его рекомендуется использовать лишь для относительно малых объемов данных. Кроме того, необходимо быть очень внимательными, чтобы получить правильное форматирование/конвертирование типов данных каждого столбца.

6.2 Использование технологии Sybase Enterprise Connect Data Access (ECDA) для Oracle

ECDA – это продукт компании Sybase, обеспечивающий связь между Oracle и Sybase ASE. Он выступает в качестве шлюза: с его помощью можно напрямую подключаться из БД ASE к БД Oracle и переносить данные Oracle непосредственно в ASE, используя лишь SQL.

Функциональность ECDA основана на «прокси-таблице» ASE, которая соотносится с актуальной таблицей Oracle. Путем выбора из прокси-таблицы запускается извлечение данных из таблицы Oracle и эти данные можно вставить непосредственно в таблицу ASE. Кроме того, поддерживается функция объединения таблиц Oracle (через их прокси-таблицы) с таблицами в Sybase ASE.

Главное преимущество ECDA заключается в том, что программа автоматически учитывает тип данных при конвертировании из Oracle в Sybase ASE. Данные продукта позволяют гибко контролировать использование языка SQL для получения доступа к прокси-таблицам.

При работе ECDA запускается отдельный процесс, выполняемый за пределами сервера ASE.

6.2.1 Пример использования ECDA

Приведем пример работы ECDA.

1. Будем считать, что инструмент ECDA установлен и запущен. Определите в ASE удаленный сервер Oracle:

```
sp_addserver ORACLEDC, direct_connect, ORACLEDC
```

В результате откроется доступ к серверу Oracle, обозначенному как «ORACLEDC».

2. В базе данных Oracle существует следующая таблица:

```
example_ora_table
(id_num int,
name varchar(30),
```

```
phone varchar(20) null,  
birthdate date null)
```

3. В базе данных Sybase ASE создайте следующую прокси-таблицу:

```
create existing table example_proxy_table  
(id_num int,  
name varchar(30),  
phone varchar(20) null,  
birthdate smalldatetime null)  
external table  
at 'ORACLEDC.oradb.example_ora_table'
```

4. Теперь с помощью SQL можно получить доступ к таблице из примера (example_proxy_table), как если бы она хранилась в базе данных Sybase ASE.

Следующий код копирует все строки из таблицы Oracle в таблицу ASE example_sybase_table:

```
insert example_sybase_table  
select * from example_ora_table
```

6.3 Использование Sybase Replication Server Heterogeneous Edition (RSHE) для Oracle

Клиенты Sybase часто используют Sybase Replication Server для облегчения миграции баз данных. Основное достоинство этой технологии – значительное сокращение времени простоя, в течение которого данные переносятся из старой БД в новую.

6.3.1 Минимальное время простоя при миграции с использованием сервера репликации

Сервер репликации Sybase извлекает транзакции БД Oracle путем чтения журналов повтора/отмены действий, а затем применяет эти транзакции к ASE, поддерживая тем самым БД ASE в актуальном состоянии. Кроме того, сервер репликации позволяет скопировать все содержимое таблиц Oracle в ASE, чтобы инициализировать репликацию данных («материализация системы репликации»). Если речь идет о больших таблицах, то следует решить главный вопрос: как проводить первоначальную реализацию – с помощью сервера репликации или посредством внешнего механизма выгрузки и загрузки.

Цель использования сервера репликации для переноса данных – полная синхронизация баз данных ASE и Oracle. В этот момент приложения можно переключить с БД Oracle на БД ASE (после перехода на новую систему настройки репликации можно удалить). Длительность простоя определяется лишь временем, необходимым для перехода приложения на новую систему, и обычно измеряется минутами, а не часами.

Важно отметить, что при репликации обеспечивается прозрачность работы приложений: приложения продолжают нормально функционировать на базе данных Oracle до наступления момента переключения на новую систему (очевидно, что потребуются модификация самих приложений для поддержки БД ASE, но мы не касаемся этого вопроса при обсуждении данного аспекта миграции данных).

Другие решения требуют гораздо больше времени для переноса данных. Это объясняется тем, что сервер репликации использует механизм инкрементной выгрузки обновленных данных из Oracle в ASE и благодаря этому приложения могут продолжать нормальную работу. Большинство же других способов миграции предполагают создание копии всей таблицы вначале (при этом необходима полная остановка работы приложений либо их переход в режим «только чтение»), поскольку очень сложно синхронизировать измененные данные с уже скопированными данными. Поэтому требуемое время простоя системы приблизительно равно времени, необходимому для копирования данных из системы Oracle в Sybase.

6.3.2 Исходная материализация для настройки репликации

Сервер репликации может автоматически копировать полное содержимое таблиц Oracle в ASE, чтобы инициализировать репликацию данных («материализация системы репликации»). Однако для очень больших таблиц этот процесс может длиться неприемлемо долго. В качестве альтернативного подхода можно использовать следующее: создать исходную копию больших таблиц с помощью других средств (см. варианты, описанные в разделе о миграции данных, например,

выгрузку данных в плоский ASCII-файл, а затем загрузку их в ASE с помощью bcp). При использовании сервера репликации вносимые в таблицу изменения будут впоследствии синхронизированы с новой системой. На верхнем уровне этот подход выглядит следующим образом:

1. Настройка репликации всех таблиц Oracle в таблицы ASE, но без установки автоматической материализации больших таблиц. При желании можно активировать «автоподключение» для больших таблиц (в зависимости от того, насколько понятен процесс изменений, вносимых в данные; более подробные сведения можно найти в документации для Replication Server).
2. Приостановка DSI-подключения сервера репликации к базе данных ASE. Впоследствии Replication Server будет отслеживать все изменения, вносимые в таблицы Oracle, и помещать их в стабильные очереди запросов. Позже эти изменения будут применены в ASE.
3. Копирование и загрузка данных в ASE для больших таблиц (с помощью выбранного метода).
4. Восстановление подключения Replication Server к базе данных ASE после завершения загрузки больших таблиц в ASE. При этом изменения, накопленные в стабильных очередях запросов Replication Server, будут применены к таблицам ASE.
5. После того как в ASE будут переданы все изменения, состояние базы данных ASE должно в точности совпадать с таковым базы данных Oracle. Теперь можно переключать приложения на новую систему – на этом процесс миграции будет завершен.

Если таблицы не слишком велики для выполнения автоматической материализации, либо если длительное время выполнения этого процесса приемлемо (поскольку приложения Oracle продолжают функционировать), то вышеуказанные шаги можно заменить простой настройкой репликации таблицы из Oracle в ASE с помощью автоматической материализации.

6.3.3 Прочие соображения

Для использования данного варианта необходимо приобрести продукт Sybase Replication Server Heterogeneous Edition (RSHE) для Oracle, а также научиться работать с ним. Копирование данных из Oracle в ASE с помощью сервера репликации не подходит для работы с большими объемами данных. В этом случае исходную материализацию системы репликации эффективнее выполнять с помощью одного из дополнительных вариантов, представленных здесь.

Прежде чем использовать Sybase Replication Server для репликации базы данных Oracle, убедитесь в том, что это не нарушает условий существующих лицензий Oracle. Если используется полная лицензия Oracle, то ограничения отсутствуют; если же лицензия Oracle ограничена (например, только на работу продукта), то использование Replication Server может быть юридически запрещено, и потребуется дополнительная лицензия Oracle. Sybase не занимается подобными вопросами, поэтому решать их надо, обратившись в Oracle.

6.4 Использование инструмента ETL стороннего разработчика, которое поддерживает и Oracle, и Sybase ASE.

Если в компании уже используется инструмент ETL, поддерживающий обе системы (Oracle и Sybase), то его можно задействовать для миграции данных. Обычно для этого требуется остановка системы на время переноса данных из Oracle в Sybase, за исключением случая, когда инструмент ETL способен сортировать изменения, вносимые в данные во время процесса переноса.

Проверьте соответствие всем лицензионным ограничениям и уточните у поставщика возможность использования данного инструмента для переноса данных из Oracle в Sybase ASE.

6.5 Типы данных Oracle, требующие особого внимания при миграции

Ниже указаны типы данных Oracle, которые требуют особого внимания при миграции данных.

- **Oracle TIMESTAMP Sybase BIGDATETIME**
грануляция типа данных TIMESTAMP у Oracle составляет 1/100000000 долю секунды. Это превышает точность типа данных BIGDATETIME у Sybase, гранулярность которого составляет 1 микросекунду. При миграции данных с помощью bcp нужно редактировать данные TIMESTAMP путем удаления последних трех цифр, чтобы в bcp не появлялось сообщение об ошибке.
- **Oracle BLOB/CLOB/NCLOB Sybase IMAGE/TEXT/UNITEXT**
В Oracle для хранения крупных двоичных объектов используется тип данных BLOB, а для хранения крупных символьных объектов – CLOB. Оба типа могут хранить до 128 Тб данных (размер блока базы данных – 4 Гб*), если

говорить о выпуске Oracle 11g. При миграции данные типа BLOB в Oracle должны соотноситься с типом данных IMAGE в Sybase, а CLOB – с типом данных TEXT. Максимальный размер для отдельного столбца типов данных IMAGE или TEXT в Sybase ASE составляет 2 Гб. Если актуальные значения данных Oracle превышают этот предел, то их нельзя сохранить в ASE. В данном случае поможет Sybase IQ, поскольку он поддерживает максимальное значение столбца от 512 Тб до 2 Пб.

- **Oracle BFILE**

Тип данных Oracle BFILE служит для хранения указателей (ссылок) на внешний двоичный файл, который хранится вне базы данных. Для данного типа нет прямого функционального эквивалента в Sybase ASE, поэтому может потребоваться внесение изменений в приложение.

7 Миграция PL/SQL в Transact-SQL

PL/SQL – это реализация языка SQL компанией Oracle. Transact-SQL (T-SQL) – это диалект SQL, реализованный в Sybase ASE. Обе версии SQL в целом совместимы со стандартом ANSI-92 на уровне ввода, однако оба разработчика внедрили множество собственных расширений и усовершенствований, не согласующихся с ANSI.

Во многих случаях оба диалекта имеют одинаковую функциональность своих фирменных расширений, однако при миграции из PL/SQL в T-SQL могут потребоваться синтаксические изменения или модификация кода (в той или иной степени). В тех случаях, если в T-SQL нет прямого эквивалента конкретной логической структуре PL/SQL, придется прибегнуть к значительной переделке кода или даже переписыванию приложения.

Хотя несовместимость между Oracle и Sybase носит ограниченный характер (если говорить о миграции схемы и миграции данных), при переходе могут возникнуть значительные сложности из-за наличия двух диалектов SQL. Поэтому миграция PL/SQL в T-SQL представляет самую сложную часть любого процесса перехода с Oracle на Sybase, и в этом случае обычно требуется, чтобы конвертирование/миграция выполнялись вручную.

Ключевой фактор для успешного перехода – реалистичная оценка сложностей, связанных с миграцией SQL, до начала этого процесса. С этой целью в главе 3 приведены контрольные перечни вопросов.

Чтобы помочь специалистам в процессе реальной миграции PL/SQL в T-SQL, в главе 11 приведены перекрестные ссылки между функциями Oracle и их эквивалентами в Sybase ASE, разбитые по сложности на три категории. Эти ссылки представляют собой расширенный вариант контрольного перечня Oracle из главы 3: в нем даны более подробные сведения и конкретные предложения о миграции той или иной функции Oracle в ASE.

7.1 Местонахождение кода PL/SQL

Код PL/SQL располагается в следующих местах:

- сохраненные процедуры (на сервере баз данных);
- триггеры (на сервере баз данных);
- SQL-функции (на сервере баз данных);
- SQL-запросы (отправленные на сервер баз данных клиентскими приложениями).

Объекты PL/SQL на сервере баз данных можно декомпилировать либо принять в качестве начальной точки скрипты репозитория, которые использовались для создания этих объектов (если эти скрипты существуют и находятся в актуальном состоянии).

Код PL/SQL, находящийся в клиентских приложениях, следует выявлять другим способом, например, путем анализа исходного кода.

Если говорить об использовании существующих скриптов или о декомпиляции объектов PL/SQL с сервера баз данных, то к ним применимы те же подходы, что и по отношению к схеме базы данных (обсуждение доводов «за» и «против» приведено в разделе 4.1).

Для декомпиляции объектов PL/SQL также можно использовать Sybase PowerDesigner (см. раздел 4.2); однако PowerDesigner не поддерживает конвертирование в T-SQL (для конвертирования оцените возможности инструмента SwisSQL, см. ниже).

Поскольку подавляющая часть PL/SQL обычно содержится в сохраненных процедурах и триггерах, миграцию PL/SQL часто приравнивают к миграции сохраненных процедур и триггеров. Хотя с формальной точки зрения такое определение неверно (как показано выше, есть и другие места, где используется PL/SQL), оно правильно очерчивает область, где возникает большинство проблем миграции.

7.2 SwisSQL – помощь при миграции PL/SQL в T-SQL

При выполнении миграций из Oracle в ASE специалисты охотно пользуются инструментами, помогающими совершать перенос из PL/SQL в T-SQL. Однако маловероятно найти инструмент, способный автоматически выполнить безошибочную миграцию всего кода PL/SQL из реальной системы Oracle в T-SQL, используемый в ASE. Тем не менее, есть надежда, что существенную часть работы можно будет сделать с помощью подходящего инструмента. Хотя надо быть готовыми к тому, что некоторые преобразования/миграцию придется выполнить в ручном режиме.

SwisSQL (инструмент стороннего разработчика) – одно из немногих средств, помогающих при автоматической миграции кода Oracle PL/SQL в Sybase T-SQL. Для получения дополнительной информации см. <http://www.swissql.com/products/oracle-to-sybase>.

Мы рекомендуем оценить возможность использования SwisSQL для миграции PL/SQL.

Согласно www.swissql.com, инструмент поддерживает следующие аспекты:

- Перенос процедур, функций, отображений и пакетов из PL/SQL в Transact-SQL. Поддерживаются как именованные, так и анонимные блоки PL/SQL.
- Переменные, курсоры, определенные в спецификации пакета и в теле пакета.
- Вызовы процедур, сделанные внутри другой процедуры.
- Перенос встроенных и пользовательских типов данных PL/SQL в Transact SQL.
- Перенос встроенных функций PL/SQL в соответствующие процедуры Transact SQL.
- Перенос встроенных пакетов, например, DBMS_OUTPUT и DBMS_SQL.
- Объявление курсора, операторы OPEN, FETCH и CLOSE, переменные курсора и неявные курсоры.
- Перенос переменных REF Cursor в Transact-SQL.
- Атрибуты курсора %ROWCOUNT, %FOUND, %NOTFOUND и %ISOPEN.
- Объекты последовательности.
- Метка PL/SQL.
- Скалярные переменные и переменные Record Anchor.
- Задаваемые пользователем исключения.
- Параметры ввода/вывода PL/SQL.
- Операторы Native Dynamic SQL (NDS).
- Тип Collection.
- Функции транзакций: COMMIT, ROLLBACK и SAVEPOINT.
- Условные и контрольные операторы.
- Механизм настройки сопоставления типов данных, который должен использоваться при миграции с помощью файла конфигурации.

8 Сравнение транзакций и блокировки в Oracle и Sybase

Наиболее существенные расхождения между Oracle и Sybase ASE наблюдаются в вопросах обработки транзакций, изоляции транзакций и блокировки. Поэтому этой теме посвящена отдельная глава.

8.1 Сравнение блокировки в Oracle MVCC и Sybase

Цель транзакций в базе данных – перевод базы данных из одного согласованного состояния в другое. При всех транзакциях в базах данных – как в Oracle, так и в Sybase ASE – гарантируется полное сохранение характеристик ACID, определяемых стандартом ANSI. ACID. Это аббревиатура от следующих слов:

- **Atomic (атомарность):** применяются либо все модификации транзакции, либо ни одной.
- **Consistent (согласованность):** транзакция переводит базу данных из одного согласованного состояния в другое при сохранении ограничений в виде целостности ссылок.
- **Isolated (изолированность):** влияние транзакции, совершенной одним пользователем, невидима для других пользователей, пока транзакция не будет совершена.
- **Durable (долговечность):** после успешного выполнения транзакции ее нельзя отменить.

Реализация компанией Oracle аспекта изолированности транзакций отличается от аналогичной в Sybase ASE. В Oracle используется MVCC (мультиверсионный контроль параллельного выполнения операций): открытая транзакция создает новую версию тех данных, которые эта транзакция модифицирует. Причем это происходит таким образом, что при обращении к этим же данным в других сессиях будет считываться немодифицированная версия данных, и блокировки не будет («писатели не блокируют читателей, читатели не блокируют писателей»). В Sybase ASE, наоборот, поддерживается единственная версия данных, а чтобы реализовать изолированность транзакции, используется блокировка.

В Oracle тоже используется блокировка, помимо MVCC, однако ее концепция радикально отличается от таковой в ASE.

8.2 Вопросы миграции, связанные с транзакциями

Различие в подходах к изоляции транзакций в Oracle и Sybase ASE может привести к появлению следующих проблем при миграции:

- В основе приложений и запросов Oracle лежит, явно или неявно, такое поведение Oracle MVCC, которое можно охарактеризовать как «писатели не блокируют читателей, читатели не блокируют писателей». При миграции подобных запросов в неизменном состоянии в Sybase ASE могут возникнуть проблемы, связанные с одновременным выполнением операций, ведущие к блокировке.
- Длительно выполняемые транзакции: они являются обычным делом в Oracle, поскольку MVCC позволяет транзакциям оставаться открытыми в течение более длительного времени с меньшим количеством побочных эффектов (хотя, естественно, здесь тоже есть определенные ограничения, например, писатели все же блокируют писателей). В случае СУБД Sybase ASE, разработанной специально для высокопроизводительных систем обработки транзакций (OLTP), транзакции должны быть максимально короткими для достижения наилучших параметров одновременной работы и высокой производительности. Длительные транзакции быстро приводят к возникновению проблем при параллельном выполнении операций (блокировке) и высокому ресурсопотреблению. Это оказывает воздействие и на транзакции других пользователей (переполнение журнала транзакций).
- Oracle использует неявные транзакции (называемые связанными транзакциями). Это означает, что транзакция запускается автоматически при выполнении оператора SELECT, INSERT, UPDATE или DELETE. По умолчанию в Sybase ASE используются явные транзакции (несвязанные), хотя эта СУБД поддерживает также соответствующие стандарту ANSI неявные и связанные транзакции. При миграции в ASE может потребоваться выполнить некоторые транзакции в режиме связанных, что связано с внесением изменений в способ обработки этих транзакций путем существенного изменения режима их выполнения на уровне сессии или клиента либо посредством добавления явных операторов BEGIN TRANSACTION (которые в Oracle не поддерживаются и не используются).

Чтобы выполнить успешную миграцию на Sybase ASE, критически важно досконально знать поведение приложений Oracle в вышеуказанных аспектах, а также основы построения запросов и обработки транзакций в этих приложениях.

8.3 Использование в ASE неявных/связанных транзакций

Наиболее простой способ миграции неявных/связанных транзакций Oracle – сохранение структуры транзакций в приложении Oracle и работа с Sybase ASE в режиме связанных транзакций.

Переход в режим неявных/связанных транзакций в Sybase ASE осуществляется следующим образом:

- Выполнение команды SET CHAINED ON на языке T-SQL перед запуском транзакции. Этот оператор можно также выполнить в ASE в триггере логина.
- Задание значения «true» для атрибута подключения OpenClient CS_OPT_CHAINXACTS (значение по умолчанию = false) в клиентском приложении до подключения к серверу ASE (при использовании утилиты isql этот атрибут задается путем указания флага командной строки «Y»).

Поскольку некоторые операции в Sybase ASE не поддерживают режим связанных транзакций, например, процедуры администрирования sp_configure, то использовать данный режим для всех подключений на практике нецелесообразно (хотя ASE 15.7 допускает выполнение различных системных процедур sp_* в режиме связанных транзакций). Рекомендуется включать режим связанных транзакций только для тех подключений или сохраненных процедур, для которых он действительно нужен. Для подключений, выполняемых с целью администрирования базы данных (обычно с логином sa), следует использовать установленный по умолчанию режим несвязанных транзакций.

8.3.1 Язык управления транзакциями DDL

При работе Sybase ASE в режиме связанных транзакций возможна ситуация, когда непосредственно в процессе миграции из Oracle внутри транзакции выполняются операторы DDL. По умолчанию в Sybase ASE появляется сообщение об ошибке. Чтобы в ASE разрешить выполнение операторов DDL, выполните команду: sp_dboption database_name, 'ddl in tran', true .

Кроме того, после каждого оператора DDL в системе Oracle появляется неявный оператор COMMIT. В ASE следует вставить явный оператор COMMIT после каждого оператора DDL, выполняемого в транзакции, чтобы избежать проблем с параллельным выполнением операций. Еще один вариант: временно отключить режим связанных транзакций на уровне сессии, пока выполняется DDL.

8.3.2 Обработка транзакций в сохраненных процедурах

Если обработка транзакций выполняется внутри сохраненных процедур и важен режим транзакций (связанный/ несвязанный), то Sybase ASE позволяет задать принудительное выполнение сохраненной процедуры только в режиме связанной или только в режиме несвязанной транзакции. Для этого используется команда sp_procxmode:

```
sp_procxmode proc_name, { 'chained' | 'unchained' | 'anymode' }
```

8.4 Использование в ASE режима явных/несвязанных транзакций

Если окажется, что запуск ASE в режиме неявных/связанных транзакций вызывает слишком большое количество проблем с операциями, выполняемыми параллельно, подумайте о возможности использования установленного в ASE по умолчанию режима явных/несвязанных транзакций для всех (или только для выбранных) транзакций.

При использовании этого режима необходимо добавлять ко всем выполняемым транзакциям оператор BEGIN TRANSACTION (этот оператор не используется в режиме связанных транзакций, которые запускаются неявно). Чтобы определить оптимальное место для добавления BEGIN TRANSACTION, необходимо четко понимать, как выполняется данная транзакция. В любом случае рекомендуется, чтобы длина транзакций в ASE была как можно более короткой.

8.5 Использование в ASE усовершенствований, обеспечивающих параллельное выполнение транзакций

Существует тенденция, особенно со стороны компании Oracle, рассматривать функциональность MVCC в системах Oracle, как самую лучшую и незаменимую модель обработки транзакций по сравнению с продуктами других разработчиков.

На практике же большую часть преимуществ параллельного выполнения операций, которые дает MVCC, можно получить в Sybase ASE благодаря использованию ее специфических функций. Однако проблемы, вызванные неоптимальной структурой транзакций и (или) запросов при параллельном выполнении операций, в Oracle проявляются медленнее, чем в ASE, следовательно, дисциплина программирования транзакций важнее для Sybase ASE.

При миграции из Oracle рекомендуется подумать об использовании в ASE следующих параметров и функций:

- Выберите схему блокировки страниц данных или строк данных для таблиц БД. Эти схемы улучшают возможности параллельного выполнения операций по сравнению со значением по умолчанию «allpages», которое следует использовать для осуществления миграции из Oracle в ASE (см. также раздел 4.9). Если используется блокировка строк данных, то незафиксированные (еще не совершенные) вставки не блокируют читателей; кроме того, в некоторых сценариях будет автоматически применяться псевдо-блокировка на уровне столбца (см. «Руководство по быстродействию и настройке», том «Блокировка и контроль параллельного выполнения операций» для получения дополнительных сведений).
- Подумайте о возможности использования в запросах аргумента «readpast». В этом случае при чтении данных запрос будет пропускать блокировки, которые могут заблокировать операцию чтения. Например:

```
select * from mytable readpast where mykey = 123
```

При использовании аргумента «readpast» страница данных (со схемой блокировки страниц данных) или строка данных (со схемой блокировки строк данных), будучи заблокирована и пропущена, читаться не будет. Однако во многих случаях это допустимо, поскольку тайминги запроса таковы, что другие пользователи не имеют доступа к заблокированным данным. Либо известно, что пропущенные данные не влияют на результат запроса.

- Можно использовать уровень изоляции транзакций 0 согласно стандарту ANSI (ANSI READ UNCOMMITTED) в запросах SELECT. Это позволит запросу SELECT считывать заблокированные в настоящий момент данные, которые, кстати, могут обновляться в это время в результате транзакции, выполняемой другим пользователем. При использовании уровня изоляции транзакций 1 согласно стандарту ANSI (READ COMMITTED) вместо этого будет заблокирован запрос SELECT.

При использовании уровня изоляции 0 согласно ANSI важно понимать возможные побочные эффекты, например: требование уникальной индексации, риск того, что в некоторых сценариях выполнение запросов SELECT с уровнем изоляции 0 будет прервано (см. «Руководство по быстродействию и настройке», том «Блокировка и контроль параллельного выполнения операций» для получения дополнительных сведений).

Кроме того, поскольку уровень изоляции 0 не обеспечивает реальной изоляции транзакций, существует риск считывания тех данных, которые в данный момент обновляются и могут вскоре еще раз обновиться либо принять исходные значения после считывания. Однако это допустимо, поскольку известно, что чтение незафиксированных данных не оказывает никакого влияния на результат запроса.

При использовании уровня изоляции транзакций 0 настоятельно рекомендуется не задавать его в качестве уровня изоляции для сессии по умолчанию, а добавить условие AT ISOLATION READ UNCOMMITTED или AT ISOLATION 0 только для тех операторов SELECT, где требуется уровень изоляции 0.

8.6 Другие аспекты выполнения транзакций

- Точки сохранения: в Sybase ASE, как в Oracle, поддерживаются точки сохранения, хотя синтаксис немного отличается.
- По умолчанию Oracle работает на уровне изоляции транзакций 1 (READ COMMITTED), как и Sybase ASE. В Oracle также поддерживается уровень изоляции транзакций 3 (SERIALIZABLE). Sybase ASE поддерживает оба этих уровня изоляции (обратите внимание на то, что в Oracle не поддерживаются уровни изоляции 0 и 2).
- Поведение SQL*Plus при фиксации:
 - o При использовании в Oracle SQL*Plus и нормальном выходе из системы всегда выполняется фиксация данных. В Sybase при существовании isql-клиента фиксация не происходит и, следовательно, происходит откат любой открытой транзакции – такое поведение противоположно поведению SQL*Plus в Oracle.
 - o SQL*Plus в Oracle можно настроить на выполнение автоматической фиксации после каждого оператора – с помощью оператора SET AUTOCOMMIT ON; по умолчанию автофиксация отключена. В Sybase isql-клиент не поддерживает автофиксацию; для получения аналогичного результата следует вставить в явном виде операторы COMMIT.
- В Oracle поддерживается синтаксис SET TRANSACTION READ ONLY, который позволяет читать данные при выполнении транзакции «только чтение данных». В результате достигается практически тот же самый эффект, что и при задании уровня изоляции транзакций 3 (SERIALIZABLE).

В Sybase ASE следует изменить указанный синтаксис и задать использование уровня изоляции транзакций 3 (SERIALIZABLE). Для этого можно выбрать любую из синтаксических структур:

- o SET TRANSACTION ISOLATION LEVEL 3
- o SELECT... AT ISOLATION LEVEL 3

Вместо ISOLATION LEVEL 3 можно также использовать ISOLATION LEVEL SERIALIZABLE.

- В Oracle поддерживается структура SET TRANSACTION READ WRITE; ее можно удалить при миграции, поскольку в Sybase ASE такое поведение транзакций установлено по умолчанию.
- Зависания. Представители Oracle иногда заявляют о том, что продукты других производителей баз данных, которые не поддерживают MVCC, являются источником «зависаний». Возможно, они используют такую устрашающую терминологию как аргумент в борьбе с конкурентами.

Действительно, зависания редко происходят в среде Oracle, хотя следует отметить (обратитесь к разделу о параллельных вычислениях в любом учебнике по теории вычислительных систем), что возможность зависания никогда нельзя полностью исключать в многопользовательской среде и данное правило распространяется и на БД Oracle.

Если руководствоваться элементарными передовыми практиками, то зависания в Sybase ASE не происходят вообще или, в крайнем случае, очень редко. Основная рекомендация, которая позволит избежать зависаний, состоит в следующем: транзакции должны обращаться к нескольким таблицам в одинаковом порядке. Кроме того, снизить вероятность зависаний поможет и применение схемы блокировки строк данных ASE. Наконец, на случай если зависание все же произойдет, рекомендуем по возможности добавить в приложение логическую структуру «retry».

В заключение отметим, что зависания – это не самая большая проблема, возникающая в ходе миграции из Oracle на Sybase ASE.

9 Прочие аспекты миграции

9.1 Курсоры

Главное различие между Oracle и Sybase ASE – способ обработки результаты запросов. В Sybase ASE результирующий набор представляет собой идентификатор, для которого используется «обработка набора». Это означает, что в сохраненной процедуре результирующие наборы хранятся во временных таблицах и впоследствии уточняются, в то время как способ Oracle основан на курсорной обработке при навигации по результирующим наборам. В основе подхода Oracle лежит поддержка версионности транзакций и обеспечение целостности данных благодаря использованию курсоров. В Sybase ASE аналогичный эффект для результирующего набора, полученного с помощью курсора, достигается благодаря использованию нечувствительных курсоров, и это наиболее полно соответствует реализации курсоров в системах Oracle.

Поддержка курсоров есть и у Oracle, и у Sybase ASE – она лишь незначительно различается по синтаксису и семантике.

Oracle PL/SQL реализован с помощью процесса неявного освобождения курсора. При закрытии курсора в Oracle происходит его автоматическое освобождение. Чтобы сделать это в Sybase ASE, требуется оператор явного освобождения курсора.

REF CURSOR – это тип данных в Oracle. С его помощью можно создавать параметры и переменные (называемые «переменными курсора»). Переменная курсора действует как указатель на результирующий набор, и ее можно связать с запросами во время выполнения, а также передавать между сохраненными процедурами, функциями и пр. Таким образом, переменную курсора можно открыть в одной сохраненной процедуре, а результаты получить в другой, то есть эта переменная передается между процедурами. Поскольку в ASE отсутствует понятие REF CURSOR, необходимо переписать код PL/SQL в тех местах, где используется REF CURSOR (например, переписать сохраненные процедуры или поместить результаты запроса во временные таблицы и предоставить доступ к этим таблицам для сохраненных процедур).

9.2 Последовательности

В Sybase ASE отсутствует полный эквивалент последовательностей Oracle, но в некоторых случаях (startwith 1 incremented by 1, значения последовательности в разных таблицах не используются совместно) это можно легко исправить с помощью столбца идентичности (identity column).

В других случаях функциональность последовательностей следует эмулировать с помощью таблицы ключей (key counter table) и сохраненной процедуры:

Код Oracle:

```
CREATE SEQUENCE test_seq
MINVALUE 1
STARTWITH 1
INCREMENTED by 1
CACHE 20;
```

```
INSERT INTO m_table VALUES (test_seq.nextval,..);
```

Соответствующий код Sybase:

```
-- create table
CREATE TABLE my_seq (seq int)
go

-- Initialize the sequence
INSERT INTO my_seq select 0
go

-- create stored procedure to increment and return the value
-- note that this can also be done with an OUTPUT parameter
CREATE PROCEDURE get_seq (@incr int)
AS
UPDATE my_seq SET seq = seq + @incr
SELECT @seq = seq FROM my_seq
RETURN @seq
go
```

```
-- execute the procedure to get the next sequence number
DECLARE @seq int
EXEC @seq = get_seq 1
INSERT INTO m_table VALUES (@seq, ...)
go
```

Другой подход состоит в том, чтобы заменить функциональность последовательностей с помощью статической Java-функции, которая видима во всех процессах (то есть она будет загружена системой ClassLoader). Этот вопрос не обсуждается подробно в данном Руководстве.

9.3 Обработка ошибок/исключений

В Oracle каждый оператор SQL автоматически проверяется на наличие ошибок, прежде чем произойдет переход к следующему оператору. Если встречается ошибка, то инструмент контроля немедленно переключается на средство обработки исключений (если оно существует). PL/SQL поддерживает создание пользовательских средств обработки исключений, чтобы обрабатывать различные типы ошибок. В Sybase ASE переход управления от одного оператора к другому происходит без проверки на наличие ошибок. Это означает, что подобную проверку здесь нужно делать после каждого оператора SQL.

В Oracle встроенная процедура RAISE_APPLICATION_ERROR уведомляет клиента о возникновении ошибки на сервере и сразу же возвращается в вызывающую подпрограмму. В начале процедуры создается неявная точка сохранения SAVEPOINT. Встроенная процедура RAISE_APPLICATION_ERROR выполняет откат до точки SAVEPOINT или до последней зафиксированной транзакции в пределах процедуры. После этого управление возвращается в вызывающую подпрограмму.

В Sybase ASE эквивалентом процедуры Oracle RAISE_APPLICATION_ERROR является RAISERROR. В отличие от Oracle, RAISERROR не возвращает управление в вызывающую подпрограмму.

Первым шагом в конвертировании является замена всех вызовов RAISE_APPLICATION_ERROR вызовами RAISERROR, после которых сразу же должен следовать оператор RETURN для эмуляции обработки исключений Oracle.

Второй шаг – обработка неявных точек сохранения SAVEPOINT, которые создаются в Oracle перед началом каждой процедуры. Если транзакция совершается внутри одной процедуры, то это относительно просто. Однако если в коде используются вложенные сохраненные процедуры, то дело усложняется и могут потребоваться дополнительные логические конструкции для управления потоком транзакций.

9.4 Ограничения на внешнее присоединение

В Sybase ASE не разрешается устанавливать другие соединения с таблицей, у которой уже имеется внешнее соединение (см. ниже пример № 1). Кроме того, для запроса с внешним соединением и ограничением для столбца из внутренней таблицы внешнего соединения результаты могут не совпадать с ожидаемыми (пример № 2). В идеальном случае, чтобы устранить необходимость в таких связях, необходимо денормализовать структуру базы данных. В Sybase ASE обычно рекомендуется использовать синтаксис для внешнего соединения в соответствии со стандартами ANSI, а не синтаксис в стиле T-SQL (*.=.*).

Oracle	Sybase ASE
Пример № 1:	Пример № 1:
SELECT DISTINCT a. id, b.name, c.desc FROM a, b, c WHERE a. id = b. id (+) and b.id2 = c.id2 (+) (or b. id = c. id2) and a. code = 1 ORDER BY b.name	SELECT a.id, b.name, c.desc FROM a, b, c WHERE a.id = b.id and b.id2 *= c.id2 and a.code = 1 UNION SELECT a.id, '' FROM a WHERE a.code = 1 and (NOT EXISTS (SELECT 'X' FROM b WHERE a.id = b.id)) ORDER BY 2

Пример № 2:

```
SELECT a.id, b.name
FROM a, b
WHERE a.id = b.id (+)
AND b.name LIKE 'Bill%'
```

Пример № 2:

```
SELECT a.id, b.name
FROM a, b
WHERE a.id *= b.id
AND b.name LIKE 'Bill%'
```

9.5 Миграция приложений JDBC/ODBC/...

Миграция данных и любого SQL-кода, которые хранятся в базе данных (например, сохраненные процедуры и триггеры), осуществляется посредством шагов, описанных в разделе 5. В этом разделе рассказывается о различных типах приложений для клиентских баз данных, которые нужно перевести с Oracle на Sybase ASE.

- Встроенное SQL-приложение
- Клиентское приложение ODBC
- Клиентское приложение JDBC
- Библиотечное приложение, зависящее от базы данных
- Приложения C
- Формы Oracle

Во всех случаях возможно преобразование одного типа приложения в другой. Например: вместо конвертации встроенного SQL-приложения Oracle в аналогичное для Sybase можно конвертировать его в клиентское приложение JDBC.

9.5.1 JDBC

Чтобы выполнить миграцию связей JDBC из Oracle на Sybase, необходимо понимать различия в управлении драйверами JDBC в этих системах. Это и будет определять подход к способу миграции JDBC.

Oracle предоставляет следующие драйверы JDBC:

- «Тонкий» драйвер (thin driver): Java-драйвер, используемый на стороне клиента без установки клиента Oracle. Применяется как с апплетами, так и с приложениями.
- OCI драйвер (Oracle Call Interface): используется на стороне клиента после установки клиента Oracle. Применяется только с приложениями.

Oracle рекомендует использовать свой «тонкий» JDBC-драйвер во всех случаях, когда имеется подключение по TCP/IP.

В Sybase ASE существует собственный JDBC-драйвер – jConnect.

9.6 Формы Oracle

Формы Oracle, компонент продукта Oracle Developer Suite, отражают подход Oracle к быстрой и эффективной разработке и построению приложений для предприятий. Приложения, основанные на формах Oracle, могут извлекать данные из баз данных Oracle и манипулировать этими данными. Маловероятно, что приложения, основанные на формах Oracle, будут хорошо (или в принципе) работать на сервере Sybase ASE.

Sybase PowerBuilder – это инструмент разработки приложений для предприятий, позволяющий создать множество типов приложений и компонентов. Он входит в группу продуктов Sybase, которые представляют собой инструментарий для разработки клиент-серверных, многоуровневых и веб-приложений.

Приложения, основанные на формах Oracle, можно переписать с помощью PowerBuilder. Большую часть функциональных возможностей, предлагаемых формами Oracle, можно также реализовать путем использования PowerBuilder вместе с Sybase ASE.

Нельзя осуществить миграцию приложений, основанных на формах Oracle, непосредственно в приложения PowerBuilder. «Форма» – это основа интерфейса пользователя (UI) в Oracle, а «окно данных» – основа UI в PowerBuilder. Оба интерфейса являются графическими и служат для представления и приема вводимых пользователем данных. Оба они могут содержать графические и неграфические элементы.

Для получения более подробной информации о PowerBuilder перейдите по ссылке <http://www.sybase.com/powerbuilder>.

10 Перекрестные ссылки для задач администрирования баз данных (DBA)

В этой главе приводятся начальные сведения относительно соотнесения задач и концепций DBA в Oracle с таковыми в ASE. Однако инструменты и методы, используемые для администрирования баз данных и мониторинга, очень сильно отличаются друг от друга и являются специфическими для каждого из типа БД. Поэтому невозможно провести сколь-нибудь точное сопоставление. Для выполнения успешной миграции очень важно иметь высокую квалификацию в области DBA.

Описание	Oracle	Sybase ASE
Домашний каталог	\$ORACLE_HOME	\$SYBASE
База данных/Экземпляр по умолчанию	\$ORACLE_SID	\$DSQUERY
Инструмент командной строки для SQL	SQL*Plus в \$ORACLE_HOME/bin/sqlplus	isql в \$SYBASE/OCS-15_0/bin/isql
Импорт/экспорт данных	<p>команда imp / exp или impdp /expdp для «насоса» данных, расположенного в \$ORACLE_HOME/bin</p> <p>В Oracle импортируются и экспортируются данные и определения DDL плюс все прочие объекты, например, определения типов, индексы, процедуры и отображения.</p> <p>Данные, экспортированные с помощью exp можно импортировать только с помощью imp</p>	<p>Для экспорта и импорта данных: команда bcp расположенная в \$SYBASE/ASE/OCS-15_0/bin</p> <p>Для определения экспорта и импорта: команда defncopy в \$SYBASE/OCS-15_0/bin</p> <p>Для декомпиляции DDL и воссоздания кода в другой среде: команда ddlgen в \$SYBASE/ASEP/bin</p>
Загрузка данных из внешних файлов	SQL*Loader – это высокоскоростной загрузчик Oracle. Он очень быстро загружает данные в систему Oracle, однако не может выгрузить данные из БД Oracle в файл	<p>команда bcp расположенная в \$SYBASE/ASE/OCS-15_0/bin</p> <p>(с помощью bcp можно также выгружать данные в файлы)</p>
Создание новой базы данных	команда dbca в \$ORACLE_HOME/bin	База данных создается с помощью Sybase Central или посредством команды SQL
Создание нового сетевого подключения	команда netca в \$ORACLE_HOME/bin	команда dsedit в \$SYBASE/OCS-15_0/bin

<p>Настройка нового сервера Oracle Enterprise Manager (OEM)</p>	<p>команда емса в \$ORACLE_HOME/bin</p>	<p>Сочетание Sybase Central с Sybase Control Center представляет собой эквивалент OEM.</p> <p>Настройка Sybase Central: выполняется автоматически при установке Sybase ASE. Является частью клиентской установки.</p> <p>Настройка Sybase Control Center: выполняется с помощью прилагаемого установщика Sybase Installer.</p>
<p>Загрузка данных в базу данных</p>	<p>SQL*LOADER в \$ORACLE_HOME/bin/sqlldr</p>	<p>команда bcp расположенная в \$SYBASE/ASE/OCs-15_0/bin используя bcp в</p>
<p>Запуск сервера баз данных</p>	<p>Ручной запуск: Start SQL*Plus as sysdba SQL> STARTUP</p> <p>Запускает экземпляр, создает базу данных и открывает базу данных.</p> <p>Запуск с помощью скрипта: команда dbstart в \$ORACLE_HOME/bin</p> <p>Обе команды используют файлы spfiles, расположенные в \$ORACLE_HOME/dbs в следующем порядке:</p> <ol style="list-style-type: none"> 1. spfile\$ORACLE_SID.ora 2. spfile.ora 3. init\$ORACLE_SID.ora 	<p>команда startserver в \$SYBASE/ASE-15_0/install startserver -f RUN_\$DSQUERY</p> <p>Файл RUN_\$DSQUERY является эквивалентом файла spfile</p>
<p>Запуск сервера резервного копирования</p>	<p>Неприменимо</p>	<p>команда startserver в \$SYBASE/ASE-15_0/install startserver -f RUN_\$DSQUERY_BS</p> <p>Файл RUN_\$DSQUERY_BS содержит параметры запуска</p>
<p>Запуск сервера мониторинга</p>	<p>команда emctl в \$ORACLE_HOME/bin emctl start dbconsole</p>	<p>команда startserver в \$SYBASE/ASE-15_0/install startserver -f RUN_\$DSQUERY_MS</p> <p>Файл RUN_\$DSQUERY_MS содержит параметры запуска</p>
<p>Показать запущенные процессы</p>	<p>Unix: ps -aef grep \$ORACLE_SID</p> <p>Windows: pslist -d oracle</p>	<p>команда showserver в \$SYBASE/ASE-15_0/install</p>

<p>Остановка сервера баз данных</p>	<p>Войти с логином в SQL*Plus и выполнить: SQL>shutdown</p> <p>Для нормального выключения SQL>shutdown immediate;</p> <p>Для немедленного выключения SQL>shutdown abort;</p> <p>Для аварийного выключения</p>	<p>Войти с логином через isql и выполнить команду: shutdown</p> <p>go</p> <p>Без ввода параметров сервер будет ждать окончания всех транзакций. Если добавить «with nowait», то все сессии будут прекращены и сервер немедленно остановлен.</p>
<p>Остановка сервера резервного копирования</p>	<p>В Oracle отсутствует понятие «сервер резервного копирования»</p>	<p>Войти с логином через isql на сервер баз данных Sybase ASE и выполнить команду: shutdown Backup_Server_name</p> <p>go</p> <p>По умолчанию система будет ожидать окончания всех текущих процессов резервного копирования. Если добавить «with nowait», то все сессии будут прекращены и сервер резервного копирования немедленно остановлен.</p>

<p>Резервное копирование базы данных</p>	<p>В Oracle используются следующие способы резервного копирования базы данных:</p> <ul style="list-style-type: none"> • команды <code>imp / exp</code>: они позволяют импортировать/экспортировать всю базу данных целиком (все данные), отдельные схемы или отдельную таблицу. • «Насос» данных (Data Pump): новая возможность импорта/экспорта, которая появилась в версии Oracle 10g. Базовая функциональность идентична старым командам <code>imp</code> и <code>exp</code> но Data Pump работает быстрее. • RMAN: диспетчер восстановления Oracle Recovery Manager (RMAN), работа которого основана на использовании командной строки, а также продукта Enterprise Manager – предпочитаемый в Oracle способ эффективного резервного копирования и восстановления базы данных Oracle. Предусмотрены различные параметры резервного копирования, некоторые из которых требуют предварительной остановки БД Oracle. 	<p>В Sybase ASE резервное копирование всегда выполняется во время работы; никакой настройки не требуется. Это та же самая функциональная возможность, которая в Oracle называется резервным копированием Archive Log, однако здесь не требуется очистки архивного файла.</p> <p>Командный дамп database выполняет резервное копирование полной базы данных (полный дамп); дамп transaction копирует только журнал транзакций, зафиксированных после предыдущего дампа (инкрементное копирование).</p> <p>Команда <code>load database</code> восстанавливает данные из полной резервной копии; <code>load transaction</code> – выполняет инкрементное копирование после восстановления резервных копий, созданных ранее.</p> <p>Для выполнения команд <code>dump/load</code> должен быть запущен сервер резервного копирования.</p>
<p>Информация о быстродействии системы</p>	<p>Динамические отображения производительности Oracle</p>	<p>Таблицы MDA</p>
<p>Сведения о схеме</p>	<p>Отображения словарей статических данных Oracle</p>	<p>Системные таблицы (каталоги) и сохраненные системные процедуры (<code>sp_*</code>)</p>

11 Перекрестные ссылки при миграции из Oracle на Sybase

В данной главе приводятся конкретные предложения по выполнению миграции функциональных возможностей Oracle на Sybase ASE. Приведенные перекрестные ссылки представляют собой расширенный вариант контрольного перечня Oracle, приведенного в главе 3. Большую часть преобразований этого типа можно выполнить с помощью текстового редактора, используя команды поиска и замены, или с помощью инструментов типа «sed».

11.1 Миграция из Oracle на Sybase ASE: категория «Простая конверсия»

Oracle	Sybase ASE («Простая конверсия»)
Подключение к схеме Oracle CONNECT user_name/password SET ROLE	Подключение к базе данных Sybase; см. также раздел 4.5 USE database_name
В конце некоторых изученных процедур в Oracle стоит «косая черта» (слеш) /	Эквивалент команды ISQL go в конце пакета операторов SQL go
Точка с запятой служит разделителем операторов в PL/SQL ;	Разделители операторов не используются; следует удалить разделители «точка с запятой», которые применяются в Oracle
Таблица DUAL в Oracle SELECT sysdate FROM DUAL	Следует полностью удалить из запросов в Sybase ASE; однако если в запросах Oracle она встречается много раз, то ее можно создать в качестве пустой таблицы в ASE; см. раздел 4.10
Имя оператора точки сохранения SET SAVEPOINT	Имя оператора точки сохранения SAVE TRAN[SACTION]
Переменная/Определения параметров; синтаксис имен DECLARE count NUMBER	В Sybase ASE имена переменной/параметра должны начинаться с символа @. В ASE максимальная длина составляет 30 байт; в Oracle можно использовать более длинные имена DECLARE @count INT
Задание значения по умолчанию в определении переменной DECLARE blood_type char(2) := '0';	Задание значения в явном виде после определения переменной DECLARE @blood_type CHAR(2) SET @blood_type = '0'
Определения с единственным ключевым словом DECLARE DECLARE V1 NUMBER(10,0); V2 CHAR(20); CURSOR mycursor IS SELECT * FROM mytable;	При определении нескольких переменных с одним ключевым словом DECLARE в ASE следует отделять их друг от друга запятой. Курсоры определяются отдельно с помощью операторов DECLARE... CURSOR DECLARE @v1 int, @v2 char(20)

<p>Определения без ключевого слова DECLARE в разделе определений для сохраненных процедур/функций</p>	<p>В Sybase ASE требуется ключевое слово DECLARE. Курсоры определяются отдельно с помощью операторов DECLARE... CURSOR</p>
<pre>CREATE PROCEDURE p AS V1 NUMBER(10,0); V2 CHAR(20); CURSOR mycursor IS SELECT * FROM mytable; BEGIN ...statements... END;</pre>	<pre>CREATE PROCEDURE p AS BEGIN DECLARE @v1 int, @v2 char(20) DECLARE mycurs CURSOR AS SELECT * FROM mytable ...statements... END go</pre>
<p>Назначение переменных myvar := выражение;</p>	<p>SET @myvar = выражение SELECT @myvar = выражение</p>
<p>Перенос данных таблицы в переменную SELECT my_col INTO my_variable FROM my_table WHERE id = 123;</p>	<p>Непосредственный выбор в переменную SELECT @my_variable = my_col FROM my_table WHERE id = 123</p>
<p>Константы</p>	<p>Следует переопределить в качестве переменных и проверить область применения (локальная или глобальная)</p>
<p>%TYPE обозначает тип данных в столбце существующей таблицы DECLARE count my_table.id%TYPE</p>	<p>Следует определить переменную в явном виде, взяв реальный тип данных столбца</p>
<p>Динамический SQL EXECUTE IMMEDIATE '...sql...';</p>	<p>В Sybase ASE следует использовать немедленное выполнение SET @cmd = '...sql...' EXECUTE (@cmd) or: EXECUTE ('...sql...')</p>

<p>Циклы с LOOP/END LOOP:</p> <pre> LOOP ...statements...; EXIT [WHEN ...condition...]; /*exitloop*/ ...statements...; /* back to top for next iteration: */ CONTINUE [WHEN ...condition...]; ...statements...; END LOOP; </pre>	<p>Конвертировать в циклы WHILE. Используемые в Oracle операторы EXIT и CONTINUE соответствуют в ASE операторам BREAK и CONTINUE, хотя в ASE с этими операторами нельзя связать условие</p> <pre> WHILE 1=1 BEGIN ...sql... ...conditional exit... END or: WHILE <condition> BEGIN ...statements... END </pre>
<p>Циклы FOR</p> <pre> FOR i IN 1..5 LOOP ...statements... END LOOP; </pre>	<p>Следует конвертировать в циклы WHILE; для реализации FOR надо использовать переменные</p> <pre> DECLARE @i int, @i_start int, @i_end int SET @i_start = 1, @i_end = 5 SET @i = @i_start WHILE @i <= @i_end BEGIN ...statements... SELECT @i = @i + 1 END </pre>
<p>Циклы CURSOR FOR</p> <pre> DECLARE CURSOR c IS .select-statement...; LOOP myvariable IN c ...statements... END LOOP; </pre>	<p>Следует конвертировать в курсор ASE</p>
<p>Операторы и метки GOTO</p> <pre> IF var1 = -1 THEN GOTO ErrorLabel; END IF; var2 := -1; <<ErrorLabel>> var2 := -9 9; </pre>	<p>Changelabel syntax from <<labelname>> tolabelname:</p> <pre> IF @var1 = -1 GOTO ErrorLabel SELECT @var2 = -1 ErrorLabel: SELECT @var2 = -99 </pre>

Синтаксис внешнего соединения Oracle	Транслируется в синтаксис внешнего соединения Sybase ASE T-SQL или в синтаксис внешнего соединения ANSI (предпочтительно). В Sybase действуют некоторые ограничения, см. раздел 9.4
<pre>// right outer join SELECT * FROM t1, t2 WHERE t1.col1 = t2.col2 (+)</pre>	<pre>SELECT * FROM t1, t2 WHERE t1.col1 =* t2.col2 (T-SQL syntax) SELECT * FROM t1 RIGHT [OUTER] JOIN t2 ON t1.col1 = t2.col2 (ANSI syntax)</pre>
<pre>//left outer join SELECT * FROM t1, t2 WHERE t1.col1(+) = t2.col2</pre>	<pre>SELECT * FROM t1, t2 WHERE t1.col1 *= t2.col2 (T-SQL syntax) SELECT * FROM t1 LEFT [OUTER] JOIN t2 ON t1.col1 = t2.col2 (ANSI syntax)</pre>
SET TRANSACTION READ WRITE	В Sybase ASE следует удалить; см. главу 8
ALTER TABLE mytable TRUNCATE PARTITION partition_name	Заменить на TRUNCATE TABLE mytable PARTITION partition_name
CREATE OR REPLACE PROCEDURE (или FUNCTION)	Заменить на DROP PROCEDURE (или FUNCTION), затем должен следовать оператор CREATE PROCEDURE (или FUNCTION)
ALTER PROCEDURE (или FUNCTION)	Заменить на DROP PROCEDURE (или FUNCTION), затем должен следовать оператор CREATE PROCEDURE (или FUNCTION)
CREATE PROCEDURE... IS...	Заменить на CREATE PROCEDURE... AS...
<p>Параметры OUT/IN OUT для сохраненной процедуры</p> <pre>CREATE PROCEDURE p (a IN number, b OUT number, c IN OUT number) IS ...</pre>	<p>В Sybase ASE поддерживаются входные и выходные + выходные параметры, но не поддерживаются выходные параметры. Кроме того, при вызове процедуры следует указать ключевое слово output</p> <pre>CREATE PROCEDURE p @a int, @b int output, @c int output AS ... EXEC p @var1, @var2 output, @var3 output</pre>
<p>Выполнение сохраненной процедуры с поименованными параметрами (параметр => значение)</p> <pre>result := proc_name(param1 => my_var, param2 => 123) ;</pre>	<p>Следует конвертировать в синтаксическую структуру ASE с поименованными параметрами:</p> <pre>EXEC @result = proc_name @param1 = @my_var, @ param2 = 123</pre>

<p>Выполнение сохраненной процедуры с позиционными параметрами (:var)</p> <pre>VAR a NUMBER; VAR b NUMBER; VAR c NUMBER; EXEC proc_name (:a, :b, :c)</pre>	<p>Следует конвертировать в синтаксическую структуру ASE с поименованными параметрами:</p> <pre>declare @a int, @b int, @c int, @return_status EXEC @return_status = proc_name @a, @b, @c</pre>
<p>Выполнение процедур В Oracle не используется ключевое слово EXEC[UTE]</p> <pre>CREATE PROCEDURE p1 AS BEGIN ...statements... END; CREATE PROCEDURE p2 AS BEGIN p1; END;</pre>	<p>В ASE ключевое слово EXEC[UTE] является обязательным (за исключением случая, когда процедура является первым оператором в пакете)</p> <pre>CREATE PROCEDURE p1 AS BEGIN ...statements... END go CREATE PROCEDURE p2 AS BEGIN EXECUTE p1 END go</pre>
<p>Определение функции SQL с ключевым словом DETERMINISTIC</p> <pre>CREATE FUNCTION f_func (p NUMBER) RETURN NUMBER DETERMINISTIC IS.</pre>	<p>В Sybase ASE следует удалить DETERMINISTIC</p>
<p>Выполнение функции SQL</p> <pre>select myfunc (123);</pre>	<p>В Sybase ASE перед именем выполняемой SQL-функции обязательно указывается имя владельца</p> <pre>select dbo.myfunc(123) select jsmith.yourfunc(456)</pre>
<p>DECLARE CURSOR имя курсора IS...</p>	<p>Следует изменить на DECLARE имя курсора INSENSITIVE CURSOR AS...</p> <p>Концепция независимого (insensitive) курсора в ASE точнее всего соответствует таковой в Oracle</p>
<p>Курсоры Oracle</p>	<p>При закрытии курсоров Oracle происходит их автоматическое освобождение.</p> <p>В ASE курсоры надо освобождать в явном виде с помощью оператора DEALLOCATE CURSOR имя курсора. Этот фрагмент следует добавить после каждого курсора CLOSE.</p>
<p>Атрибут курсора %ISOPEN</p>	<p>Не имеет соответствия в ASE, поэтому удалите его</p>

Атрибуты курсоров %FOUND, %NOTFOUND	Конвертируйте, используя @@sqlstatus
Атрибут курсора %ROWCOUNT	Конвертируйте, используя @@rowcount
Триггеры AFTER (на уровне операторов)	Аналогично триггерам в Sybase ASE
SQL%ROWCOUNT Указывает количество строк, на которые действует последний выполненный оператор PL/SQL SELECT * FROM mytable WHERE id = 1234; IF SQL%ROWCOUNT = 0 THEN DBMS_OUTPUT.PUT_LINE ('No rows found.');	Замените на @@rowcount DECLARE @rc INT, @err INT SELECT * FROM mytable WHERE id = 1234 SELECT @rc = @@rowcount, @err = @@error IF @rc = 0 print 'No rows found.'
Тип данных BOOLEAN (только для переменных PL/SQL) Допустимыми значениями являются TRUE, FALSE и NULL	Конвертируйте в переменные типа BIT (можно использовать только 0 и 1; NULL = 0) или TINYINT NULL. Разрешается принять стандартное обозначение: 0=false; 1=true. Вместо того чтобы использовать символьные константы 0 и 1 при тестировании и задании значений, можно определить переменные с именами @true и @false (и присвоить им значения 1 и 0) и затем использовать эти имена.
Оператор MERGE	Выполняйте миграцию на версию ASE 15.7, в которой поддерживается MERGE
Разделенные таблицы с композитным разделением CREATE TABLE mytable (...columns...) PARTITION BY RANGE(ptn_key_col) SUBPARTITION BY HASH(subptn_key_col) [...]	В ASE поддерживаются разделенные таблицы, но без композитного разделения. Уберите условие SUBPARTITION
Типы данных native PL/SQL для оптимизации производительности (только для переменных PL/SQL) BINARY_INTEGER BINARY_DOUBLE BINARY_FLOAT	Конвертируйте в типы данных INTEGER, DOUBLE, FLOAT
IF-THEN-ELSE IF выражение THEN statement; ELSE statement; END IF;	В Sybase ASE отсутствуют операторы THEN или END IF, поэтому удалите их IF выражение statement ELSE statement

<p>Множественные операторы в операциях перехода IF-THEN-ELSE</p> <p>IF выражение</p> <p>THEN</p> <p style="padding-left: 20px;">statement;</p> <p style="padding-left: 20px;">statement;</p> <p>ELSE</p> <p style="padding-left: 20px;">statement;</p> <p style="padding-left: 20px;">statement;</p> <p>END IF;</p>	<p>В Sybase ASE в каждой операции перехода можно использовать только одно операторное выражение; несколько операторов необходимо сгруппировать в блок BEGIN-END</p> <p>IF выражение</p> <p>BEGIN</p> <p style="padding-left: 20px;">statement</p> <p style="padding-left: 20px;">statement</p> <p>END</p> <p>ELSE</p> <p>BEGIN</p> <p style="padding-left: 20px;">statement</p> <p style="padding-left: 20px;">statement</p> <p>END</p>
<p>Условное тестирование на основе подзапроса EXISTS</p> <p>DECLARE</p> <p style="padding-left: 20px;">v_x NUMBER (10,0);</p> <p style="padding-left: 20px;">v_temp NUMBER (1, 0) := 0;</p> <p>SELECT 1 INTO v_temp</p> <p>FROM DUAL</p> <p>WHERE EXISTS (...subquery...);</p> <p>IF v_temp = 1 THEN</p> <p style="padding-left: 20px;">v_x := -1;</p>	<p>Можно оставить в ASE в том же виде либо существенно упростить:</p> <p>DECLARE @x int</p> <p>SET @x = 0</p> <p>IF EXISTS (...subquery...)</p> <p style="padding-left: 20px;">SET @x = -1</p> <p>END</p>
<p>Оператор конкатенации строк: </p>	<p>В Sybase ASE поддерживается знак + как оператор конкатенации строк, а также знак , хотя в официальной документации это не указано</p>
<p>userenv('sessionid')</p>	<p>Эквивалентно глобальной переменной для конкретной сессии @@spid (поскольку значения @@spid используются повторно, то можно также применять значение sysprocesses.kpid, чтобы оно было уникальным)</p>
<p>MOD (X,Y)</p>	<p>X % Y</p>
<p>CEIL ()</p>	<p>CEILING ()</p>
<p>TRUNC (число)</p>	<p>CONVERT (INT,..)</p>
<p>SUBSTR ()</p>	<p>SUBSTRING ()</p>
<p>Функция SUBSTR () с двумя параметрами</p>	<p>Следует переписать, введя длину выражения в качестве третьего параметра</p>
<p>SELECT substr ('John', 2) → returns 'ohn'</p>	<p>SELECT substring ('John', 2,len ('John'))</p>
<p>LENGTH ()</p>	<p>CHAR_LENGTH () or LEN () or DATALENGTH ()</p>

CHR ()	CHAR ()
TO_CHAR(выражение)	CONVERT(VARCHAR(n), выражение)
TO_CHAR (выражение, часть даты)	Конвертируйте, чтобы можно было использовать в ASE функцию datepart()
TO_CHAR (sysdate, 'dd')	CONVERT (VARCHAR,datepart (dd,getdate ()))
TO_CHAR (выражение, строка, задающая формат)	Следует задать форматирование в явном виде, используя функции ASE
TO_CHAR (some-number, '999D99')	CONVERT (VARCHAR, ROUND (some-number, 2))
TO_CHAR (some-number, '999')	CONVERT (VARCHAR,CONVERT (INT,some- number))
TO_NUMBER (выражение)	CONVERT ([BIG SMALL TINY]INT, выражение) CONVERT (NUMERIC(p,m), выражение)
Функции и расчеты даты/времени	Перепишите, используя функции даты/времени, применяемые в ASE, например: DATEADD (), DATEDIFF (), DATEPART (), DATENAME ()
SELECT add_months (xyz ,3) FROM dual SELECT nr_days := DateEnd - DateStart	SELECT DATEADD (month, 3, xyz) SELECT @nr_days = datediff (dd, DateStart, DateEnd)
SYSDATE, SYSTIMESTAMP	Замените на GETDATE ()
TRUNC (дата/время [,единица])	Замените на CONVERT (), используя стили форматирования даты/времени
LAST_DAY ()	Функция, задающая последний день месяца и основанная на значении даты; перепишите ее, используя функции ASE SQL
Функция NVL () NVL (a,b)	Замените на функцию ISNULL () ISNULL (a,b)
Использование для идентификаторов символов верхнего и нижнего регистра (идентификаторы Oracle нечувствительные к состоянию регистра)	Следует использовать в ASE либо нечувствительный к состоянию регистра порядок сортировки, либо ввести единообразное написание идентификаторов (см. раздел 5.2)
Идентификаторы, которые являются зарезервированными словами в Sybase ASE (см. раздел 4.8)	Измените эти идентификаторы, чтобы они не совпадали с зарезервированными словами
Функция INSTR () с двумя параметрами SELECT INSTR('abcabc', 'ab')	Замените на charindex () SELECT CHARINDEX ('abcabc', 'ab')

<p>Производные таблицы (называемые также inline views – «встроенные представления») без соотнесения имени</p> <pre>select a from (select b as a, d as b from mytab) where b > 0</pre>	<p>В ASE всегда требуется указывать соответствующее имя для производной таблицы</p> <pre>select a from (select b as a, d as b from mytab) [as] somename where b > 0</pre>
<p>ALTER TABLE... SPLIT PARTITION ALTER TABLE... MERGE PARTITIONS</p>	<p>В ASE поддерживаются разделенные таблицы, но отсутствует поддержка разбиения/слияния частей. Удалите соответствующие операторы.</p>
<p>Подсказки Oracle в виде специального комментария</p> <p>За оператором SELECT:</p> <pre>SELECT /*+ INDEX (C) */ NAME FROM CUSTOMERS C WHERE ZIPCODE = 54321</pre> <p>Ключевые слова подсказок:</p>	<p>удалите все подсказки Oracle.</p>

ALL_ROWS	CURSOR_SHARING_EXACT	NOAPPEND
APPEND	DRIVING_SITE	NOCACHE
CACHE	DYNAMIC_SAMPLING	NOPARALLEL
CLUSTER	MODEL_MIN_ANALYSIS	NOREWRITE
FACT	NATIVE_FULL_OUTER_JOIN	OPT_PARAM
FIRST_ROWS	NO_NATIVE_FULL_OUTER_JOIN	ORDERED
FULL	NO_PARALLEL	PARALLEL
HASH	NO_PARALLEL_INDEX	PARALLEL_INDEX
INDEX	NO_PUSH_PRED	PQ_DISTRIBUTE
INDEX_ASC	NO_PUSH_SUBQ	PUSH_PRED
INDEX_DESC	NO_PX_JOIN_FILTER	PUSH_SUBQ
INDEX_FFS	NO_QUERY_TRANSFORMATION	PX_JOIN_FILTER
INDEX_JOIN	NO_RESULT_CACHE	QB_NAME
INDEX_SS	NO_REWRITE	RESULT_CACHE
LEADING	NO_STAR_TRANSFORMATION	INDEX_COMBINE
MERGE	NO_UNNEST	INDEX_SS_ASC
MONITOR	NO_USE_HASH	INDEX_SS_DESC
NO_EXPAND	NO_USE_MERGE	NO_INDEX_FFS
NO_FACT	NO_USE_NL	NO_INDEX_SS
NO_INDEX	NO_XML_QUERY_REWRITE	NO_MERGE
REWRITE	NO_XMLINDEX_REWRITE	NO_MONITOR
UNNEST	STAR_TRANSFORMATION	USE_HASH
USE_CONCAT	USE_NL_WITH_INDEX	USE_MERGE
	NOPARALLEL_INDEX	USE_NL

11.2 Миграция из Oracle на Sybase ASE: категория «Частичное переписывание»

Для приведенных ниже функций Oracle возможна миграция с получением частично эквивалентных функций Sybase ASE, хотя при этом может потребоваться значительное изменение синтаксической структуры и частичное переписывание алгоритмов.

Oracle	Sybase ASE («Частичное переписывание»)
<p>Указатели связей с базами данных</p> <pre>CREATE DATABASE LINK SALES.PROD [CONNECT TO CURRENT_USER] using 'SALES'; SELECT * FROM salesdata@SALES;</pre>	<p>Эквивалентно прокси-таблицам ASE, соотнесение с удаленной таблицей</p> <pre>create proxy_table sales_proxy at SALES.salesdb..salesdata select * from your_proxy</pre>
<p>Внешние таблицы</p> <pre>create table my_external_tab (... columns...) organization external (default directory external_data_dir access parameters (records delimited by newline fields terminated by '' location ('...pathname...'))</pre>	<p>Эквивалентно прокси-таблицам ASE, соотнесение с файлом O/S</p> <pre>create proxy_table my_external_tab (... columns...) external file at '...pathname...' column delimiter ''</pre>
<p>Последовательности</p> <p>Создание уникальных номеров, например для первичных ключей</p>	<p>В некоторых случаях их можно заменить, используя столбцы идентичности ASE. В остальных случаях функциональность последовательностей следует эмулировать с помощью таблицы ключей (key counter table) и сохраненной процедуры. См. раздел 9.2.</p>
<p>Функции SQL, задаваемые пользователем в виде табличных значений</p>	<p>В ASE поддерживаются только скалярные SQL-функции, задаваемые пользователем. Перепишите, используя временные таблицы.</p>
<p>Синонимы</p>	<p>Для получения синонимов таблиц или отображений замените их отображением ASE; для получения синонимов таблиц или отображений для связи с базами данных замените их прокси-таблицами ASE.</p> <p>Для получения синонимов сохраненных процедур или функций замените их процедурами/функциями, сохраненными в обертке; чтобы получить синонимы сохраненных процедур для связи с базами данных, замените их вызовом удаленной процедуры.</p> <p>Для получения других синонимов потребуется изменить приложение.</p>
<p>Комментарии для объектов базы данных</p> <pre>COMMENT ON TABLE mytab IS «This is my table»;</pre>	<p>Прямой эквивалент отсутствует. Способ сохранения комментариев к объекту в ASE описан в http://www.sybase.com/detail?id=607.</p>
<p>Индексы растровых отображений</p> <pre>CREATE BITMAP INDEX my_ix ON mytable(.</pre>	<p>ASE не поддерживает индексы растровых изображений. Удалите BITMAP и создайте обычный индекс. Растровые индексы поддерживаются в Sybase IQ.</p>

<p>Временные таблицы</p>	<p>Замените временными таблицами ASE, имена которых начинаются с символа #</p>
<pre>CREATE GLOBAL TEMPORARY TABLE temptab [...] [...]</pre>	<pre>CREATE TABLE #temptab [...] SELECT * INTO #temptab FROM my_table</pre>
<p>Вложенные таблицы</p> <p>Допустимо определение столбца как таблицы, содержащей N строк (=N значений столбца)</p>	<p>Следует заменить использование отдельной таблицы на таблицу, вложенную в столбец, задав связь первичного ключа с внешним</p>
<pre>CREATE TYPE address_t AS OBJECT (street VARCHAR2(30), city VARCHAR2 (20), state CHAR(2), zip CHAR (10)); country CHAR (30), / CREATE TYPE address_tab IS TABLE OF address_t; / CREATE TABLE customers (custid NUMBER, address address_tab) NESTED TABLE address STORE AS customer_addresses; INSERT INTO customers VALUES (654, address_tab(address_t('148 Oak Drive', 'Dallas', 'TX', '75240'), address_t('561 Virginia Road', 'Concord', 'MA', '01742'))</pre>	
<p>Таблицы объектов</p> <pre>CREATE TYPE person_type AS OBJECT (name VARCHAR2(30), address VARCHAR2(100)); CREATE TABLE person_obj_table OF person_type;</pre>	<p>Следует либо заменить обычными таблицами и столбцами, либо использовать классы Java для определения столбца как сложного типа данных, содержащего различные поля.</p>
<p>%ROWTYPE определяет запись PL/SQL с теми же самыми столбцами как особую таблицу.</p> <pre>DECLARE cust customer%ROWTYPE</pre>	<p>Определите каждое поле как отдельную переменную и измените все ссылки соответствующим образом. Либо можно конвертировать запись в Java-объект с другими элементами данных, которые могут храниться в столбце таблицы.</p>
<p>Определение типа записи PL/SQL путем нумерации полей с помощью IS RECORD OF или TYPE...IS RECORD</p>	<p>Определите каждое поле как отдельную переменную и измените все ссылки соответствующим образом. Другим вариантом является конвертирование записи в Java-объект, который будет обрабатываться как массив.</p>

<p>IS TABLE OF определяет в PL/SQL элемент «таблица» (= таблица или массив, не относящиеся к базе данных)</p>	<p>Перепишите алгоритм, используя временную таблицу, затем используйте операторы FETCH или SELECT для обработки строк. Либо можно конвертировать таблицу в Java-объект с другими элементами данных, которые могут храниться в столбце таблицы.</p>
<p>Нецелочисленное значение RETURN в сохраненной процедуре Сохраненные процедуры Oracle могут возвращать скалярные значения для любого типа данных</p>	<p>Сохраненные процедуры Sybase ASE могут возвращать только целочисленные значения. Если процедура возвращает другой тип данных, то ее следует переписать, используя выходной параметр или SQL-функцию.</p>
<p>Задаваемые пользователем пакеты</p>	<p>Переведите пакеты в индивидуальные объекты, из которых состоят пакеты (сохраненные процедуры, типы данных и так далее)</p>
<p>Переполнение в сохраненных процедурах (несколько процедур с одинаковыми именами, но разными типами данных параметров или разным числом параметров)</p>	<p>Переведите в отдельные сохраненные процедуры или разбейте на несколько сохраненных процедур</p>
<p>Обработка исключений PL/SQL; определение средств обработки исключений</p> <pre>EXCEPTION WHEN ZERO_DIVIDE THEN -- handles 'division by zero' error [...] EXCEPTION WHEN TOO_MANY_ROWS -- handle case that > 1 row affected [...] EXCEPTION WHEN NO_DATA_FOUND -- handle case that > 1 row affected [...] EXCEPTION WHEN DUP_VAL_ON_INDEX -- handle case for duplicate index key [...] (etc... other conditions exist</pre>	<p>Перепишите и проверьте с помощью @@error и @@rowcount после каждого оператора SQL</p>
<p>SQLCODE, SQLERRM Указывает статус ошибки и текст сообщения об ошибке для наиболее часто выполняемых операторов PL/SQL; используется совместно с разделом обработки исключений</p> <pre>EXCEPTION WHEN OTHERS THEN error_code := SQLCODE; error_msg := substr (SQLERRM, 1, 200); INSERT INTO audit_table (err_no,err_msg) VALUES (error_code, error_msg); END;</pre>	<p>Замените SQLCODE на @@error. В ASE нет эквивалента для SQLERRM</p> <pre>DECLARE @rc INT, @err INT SELECT * FROM mytable WHERE id = 1234 SELECT @rc = @@rowcount, @err = @@error IF @err = 0 BEGIN INSERT INTO audit_table (err_no,err_msg) VALUES (@err, ''); END</pre>

<p>RAISE_APPLICATION_ERROR</p> <p>В сохраненных процедурах выполняет откат до неявной точки сохранения в начале процедуры (или после последней зафиксированной транзакции в процедуре)</p>	<p>Следует переписать код, используя функции Sybase ASE, например, RAISERROR или PRINT, за которыми сразу же должен следовать оператор RETURN в сохраненных процедурах</p>
<p>Кодирование столбцов</p>	<p>Следует переписать, используя кодирование столбцов ASE</p>
<p>Локаторы LOB</p>	<p>Выполняйте миграцию на версию ASE 15.7, в которой поддерживаются локаторы LOB</p>
<p>Сжатие данных</p>	<p>Выполняйте миграцию на версию ASE 15.7, в которой поддерживается сжатие данных</p>
<p>Вызовы пакетов DBMS_*</p>	<p>Перепишите код, используя функциональность Sybase ASE.</p>
<p>DBMS_OUTPUT.PUT_LINE</p>	<p>PRINT</p>
<p>Извлечение данных из сохраненных процедур в приложение-клиент с использованием пакета DBMS_OUTPUT</p>	<p>Замените непосредственно оператором SELECT в сохраненной процедуре</p>
<p>Загрузчик SQL*Loader (sqlldr)</p> <p>Высокоскоростная утилита загрузки данных Oracle (только загрузка, выгрузки нет)</p>	<p>Перепишите, используя утилиту Sybase bcp</p>
<p>Глобальные переменные (в пакете PL/SQL)</p>	<p>Глобальные переменные не поддерживаются; либо проведите все переменные как параметры, либо сохраните значения в таблице и по мере необходимости выполняйте чтение/обновление этой таблицы. Другой вариант – использование статических классов Java.</p>
<p>Конструкция INTERSECT</p> <p>SELECT a FROM tab1 WHERE b > 10</p> <p>INTERSECT</p> <p>SELECT c FROM tab2 WHERE d = 0</p>	<p>Перепишите как соединение</p> <p>SELECT tab1.a FROM tab1, tab2</p> <p>WHERE tab1.a = tab2.c</p> <p>AND tab1.b > 10 AND tab2.d = 0</p>
<p>Конструкция MINUS</p> <p>SELECT a,b,c FROM tab1 WHERE...</p> <p>MINUS</p> <p>SELECT d,e,f FROM tab2 WHERE...</p>	<p>Перепишите, используя NOT IN (единственный столбец) или NOT EXISTS (несколько столбцов)</p> <p>SELECT a,b,c FROM tab1</p> <p>WHERE NOT EXISTS</p> <p>(SELECT * from tab2</p> <p>WHERE tab2.d = tab1.a</p> <p>AND tab2.e = tab1.b</p> <p>AND tab2.f = tab1.c)</p>

<p>Операторы, используемые в SQL</p>	<table border="0"> <tr><td>AS OF</td><td>CROSS</td><td>IGNORE</td></tr> <tr><td>AS OF TIMESTAMP</td><td>CUBE</td><td>ITERATE</td></tr> <tr><td>CONNECT BY</td><td>FOR</td><td>NATURAL</td></tr> <tr><td>DIMENSION</td><td>KEEP</td><td>NULLS</td></tr> <tr><td>DIMENSION BY</td><td>MAIN</td><td>NULLS FIRST</td></tr> <tr><td>EXCLUDE</td><td>MODEL</td><td>NULLS LAST</td></tr> <tr><td>GROUPING SETS</td><td>NAV</td><td>ROLLUP</td></tr> <tr><td>INCLUDE</td><td>NOCYCLE</td><td>SIBLINGS</td></tr> <tr><td>MEASURES</td><td>NOWAIT</td><td>SINGLE</td></tr> <tr><td>RETURN ALL ROWS</td><td>ON</td><td>REFERENCE</td></tr> <tr><td>RETURN UPDATED</td><td>ONLY</td><td>LOCKED</td></tr> <tr><td>ROWS</td><td>RULES</td><td>START WITH</td></tr> <tr><td>PARTITION BY</td><td>SAMPLE</td><td>UNIQUE</td></tr> <tr><td>REFERENCE</td><td>SEED</td><td>UNPIVOT</td></tr> <tr><td>SYSTIMESTAMP</td><td>SKIP</td><td>WAIT</td></tr> </table>	AS OF	CROSS	IGNORE	AS OF TIMESTAMP	CUBE	ITERATE	CONNECT BY	FOR	NATURAL	DIMENSION	KEEP	NULLS	DIMENSION BY	MAIN	NULLS FIRST	EXCLUDE	MODEL	NULLS LAST	GROUPING SETS	NAV	ROLLUP	INCLUDE	NOCYCLE	SIBLINGS	MEASURES	NOWAIT	SINGLE	RETURN ALL ROWS	ON	REFERENCE	RETURN UPDATED	ONLY	LOCKED	ROWS	RULES	START WITH	PARTITION BY	SAMPLE	UNIQUE	REFERENCE	SEED	UNPIVOT	SYSTIMESTAMP	SKIP	WAIT	<p>Следует заменить соответствующими функциональными возможностями Sybase ASE, если они есть. В противном случае придется переписать SQL</p>
AS OF	CROSS	IGNORE																																													
AS OF TIMESTAMP	CUBE	ITERATE																																													
CONNECT BY	FOR	NATURAL																																													
DIMENSION	KEEP	NULLS																																													
DIMENSION BY	MAIN	NULLS FIRST																																													
EXCLUDE	MODEL	NULLS LAST																																													
GROUPING SETS	NAV	ROLLUP																																													
INCLUDE	NOCYCLE	SIBLINGS																																													
MEASURES	NOWAIT	SINGLE																																													
RETURN ALL ROWS	ON	REFERENCE																																													
RETURN UPDATED	ONLY	LOCKED																																													
ROWS	RULES	START WITH																																													
PARTITION BY	SAMPLE	UNIQUE																																													
REFERENCE	SEED	UNPIVOT																																													
SYSTIMESTAMP	SKIP	WAIT																																													
<p>Заглавная первая буква каждого слова в строке INITCAP (строчное выражение)</p>	<p>Перепишите, используя сохраненные процедуры или SQL-функции</p>																																														
<p>Функция INSTR () с тремя или четырьмя параметрами (3=начальное положение, 4=п-ное вхождение) SELECT INSTR('abcabc', 'ab', 2) SELECT INSTR ('abcabcabc' , 'ab', 2, 2)</p>	<p>Создайте SQL-функцию для выполнения расширенного поиска. Отметим, что charindex () принимает значение 3-го параметра в ASE 15.7, но не может быть отрицательным (в случае обратного поиска), как это допускается в Oracle.</p>																																														
<p>Функция NVL2 () SELECT NVL2 (salary, salary*2,0) FROM...</p>	<p>Замените выражением CASE SELECT CASE WHEN salary = NULL THEN 0 ELSE salary * 2 END FROM...</p>																																														
<p>Функция DECODE () Используется для оценки с помощью логической конструкции типа «if-else» SELECT DECODE (T1.C1, 'ABC', T1.C2, T1.C3) as P_ID FROM T1</p>	<p>Преобразуйте в выражение CASE SELECT case T1.C1 when 'ABC' then T2.C2 else T3.C3 end as P_ID FROM T1</p>																																														
<p>Первичный ключ и внешний ключ с разными типами данных, разной точностью/масштабом (для числовых типов данных) или разной длины (для символьных типов данных)</p>	<p>В отличие от Oracle, в ASE требуется, чтобы внешний и первичный ключи имели одинаковый тип данных. Следует изменить типы данных соответствующим образом.</p>																																														
<p>Длина переменных VARCHAR2 более 16 384 байт DECLARE msg VARCHAR2(32767) (для столбцов VARCHAR2 не может быть длиннее 4000 байт)</p>	<p>Максимальная длина строчных переменных – 16 384 байт; в ASE 15.7 можно переписать код, используя более длинные строчные переменные с помощью переменных LOB или локаторов LOB</p>																																														

<p>Функции SQL, в которых последним оператором не является RETURN</p>	<p>В ASE требуется, чтобы последним оператором SQL-функции был оператор RETURN. Поэтому может потребоваться переписывание кода в части управления потоком данных.</p>
<p>Производные таблицы (называемые также inline views – «встроенные представления») с использованием синтаксиса «with» with x as (select b as a, d as b from mytab) select a from x where b > 0</p>	<p>Перепишите, используя синтаксис для производных таблиц ASE</p>
<p>UNION в курсорах</p>	<p>Курсор, содержащий UNION, нельзя перевести в ASE; такой код придется переписать</p>
<p>Директивы PRAGMA</p>	<p>Перепишите, используя синтаксис/функциональные возможности ASE</p>
<p>ROWID В таблицах Oracle всегда есть столбец ROWID с уникальным идентификатором для каждой строки, даже если для таблицы не определены первичные ключи</p>	<p>Аналогичный эффект можно получить, если в каждую таблицу добавить столбец идентичности и назвать его ROWID. В каждой таблице может быть только один столбец идентичности. Если для первичного ключа уже есть столбец идентичности, например для замены последовательности Oracle, то добавьте виртуальный вычисленный столбец с именем ROWID, равный столбцу идентичности. Этот способ подходит также, если в коде Oracle используется другое написание, например «rowid»: CREATE TABLE mytab (ROWID NUMERIC IDENTITY, rowid AS ROWID, ...другие столбцы...)</p>

ROWNUM

Для каждого столбца, возвращаемого запросом, псевдостолбец ROWNUM возвращает число, указывающее на расположение столбца в результирующем наборе. Это можно использовать в запросах, например для выбора поднабора из результирующего набора

```
SELECT * FROM emp
WHERE state = 'CA'
AND ROWNUM > 9 AND ROWNUM < 21
ORDER BYlast_name;
```

Если надо выбрать N верхних строк, то это можно сделать с помощью выражения «select top N... from...»

Если требуется более сложный выбор (например, надо получить только строки 10–20), то к результирующему набору можно добавить столбец идентичности (с именем ROWNUM) с помощью функции идентичности identity (), которая задает последовательный номер каждой строке результирующего набора. Затем этот столбец можно использовать в запросах. Обратите внимание, что для этого требуется один дополнительный шаг при создании запроса:

```
SELECT *, ROWNUM=identity (int)
INTO #t FROM emp
WHERE state = 'CA'
ORDER BYlast_name

SELECT <all columns except ROWNUM>
FROM #t
WHERE ROWNUM > 9 AND ROWNUM < 21
```

11.3 Миграция из Oracle на Sybase ASE: категория «Полное переписывание»

Для перечисленных ниже функций Oracle не существует прямых эквивалентов в Sybase ASE. Поэтому потребуется переписывание или повторная разработка алгоритмов или отдельных частей приложений.

Oracle	Sybase ASE («Полное переписывание»)
<p>Oracle MVCC (мультиверсионный контроль параллельного выполнения операций; «писатели не блокируют читателей, читатели не блокируют писателей»).</p> <p>Аспекты, представляющие интерес:</p> <ul style="list-style-type: none"> • Приложения или запросы, предполагающие отсутствие блокировки MVCC • Длительно выполняемые транзакции • DDL в транзакциях • SET TRANSACTION READ ONLY • SQL*Plus – автоматическая фиксация транзакции/ фиксация при выходе 	<p>Прямой эквивалент MVCC отсутствует. В некоторых случаях можно использовать блокировку DATAROWS, параметр READPAST или выполнять запросы SELECT, задавая уровень изоляции 0 (READ UNCOMMITTED).</p> <p>В других случаях может потребоваться изменить приложение, например, сделав транзакции как можно короче.</p> <p>См. раздел 8 для получения дополнительной информации</p>
SQL*Plus	<p>Эквивалентом SQL*Plus в Sybase ASE служит утилита isql. SQL*Plus допускает более сложные настройки конфигурации и специфические (то есть не являющиеся командами PL/SQL) команды со стороны клиента. Необходимо переписывать существующую функциональность SQL*Plus для ASE.</p>

Триггеры BEFORE	<p>Прямой эквивалент отсутствует. Некоторые аспекты функциональности (например, проверку целостности домена) можно реализовать с помощью правил или ограничений CHECK на уровне определения таблицы; однако триггер Oracle BEFORE способен выполнять намного более сложную обработку, чем с помощью правил или ограничений. Если нужную функциональность нельзя реализовать с помощью триггеров «after» в Sybase ASE, то приложение придется изменить, чтобы реализовать эту функциональность другим способом.</p>
Триггеры уровня строки (BEFORE и AFTER)	<p>Прямого эквивалента нет; см. выше информацию о триггерах BEFORE</p>
Несколько триггеров для типа DML в таблице	<p>Прямого эквивалента нет; если нужную функциональность нельзя консолидировать в одном триггере ASE, то придется изменить приложение, чтобы реализовать функциональность другим способом</p>
Реализованные отображения	<p>Прямой эквивалент отсутствует. В настоящее время Oracle использует реализованные отображения для односторонней репликации с помощью подключения к базе данных и обновляемые реализованные отображения для двухсторонней репликации.</p> <p>При необходимости подобной репликации можно использовать сервер репликации Sybase Replication Server независимо от сервера Sybase ASE.</p>
REF CURSOR	<p>Прямой эквивалент отсутствует. В ASE необходимо переписать курсоры REF CURSOR. Это можно сделать, например, путем переписывания всех необходимых сохраненных процедур или посредством помещения результатов запроса во временные таблицы с предоставлением доступа к этим таблицам различным сохраненным процедурам. См. также раздел 9.1.</p>
<p>Запросы в многооконном режиме (SELECT... OVER (...))</p> <p>SELECT name, salary, NTILE (4) OVER (ORDER BY salary DESC) AS quartile</p> <p>FROM emp</p> <p>WHERE dept_id = 123;</p>	<p>Нет соответствия в Sybase ASE. Перепишите, используя классические функции ASE; возможно, для этого придется разбить запрос на несколько этапов. С другой стороны, можно использовать Sybase IQ, который не поддерживает запросы в многооконном режиме, как и многие аналитические функции.</p>
Параметры OUT/IN OUT для функции SQL	<p>В Sybase ASE поддерживаются только входные параметры для функций SQL. Если используются выходные параметры, перепишите с использованием сохраненных процедур.</p>

<p>Недетерминированные функции SQL (функции, результат применения которых не зависит от вводимых в функцию параметров)</p>	<p>В Sybase ASE поддерживаются только детерминированные функции SQL. В функции SQL нельзя использовать DML, DDL, вызовы процедур, немедленное выполнение, вызовы некоторых функций и команд утилит. Если все это встречается в функции Oracle SQL, то перепишите код, используя сохраненные процедуры ASE.</p>
<p>Агрегатные функции SQL</p> <pre>CREATE FUNCTION f_aggr (p NUMBER) RETURN NUMBER AGGREGATE USING object-type;</pre>	<p>Прямой эквивалент отсутствует. Перепишите, используя существующие функции ASE.</p>
<p>Тип данных BFILE</p> <p>В столбце BFILE хранится указатель (ссылка) на двоичный файл за пределами базы данных</p>	<p>Прямой эквивалент отсутствует</p>
<p>Средства Oracle Streams; Oracle Data Guard</p>	<p>Используйте Sybase Replication Server</p>
<p>Oracle RAC для обеспечения высокой доступности</p>	<p>Используйте Sybase ASE Cluster Edition</p>
<p>Технология Oracle Flashback</p>	<p>Прямой эквивалент отсутствует. В некоторых случаях можно использовать существующую в ASE функцию аудита или архивные базы данных ASE, или конструкцию «until_time» при загрузке дампа журнала транзакций. В других случаях можно оставить старые значения данных, применив триггеры.</p>
<p>Псевдостолбцы Oracle, связанные с технологией Flash-back</p> <pre>ORA_ROWSCN, VERSION_XID, VERSION_STARTSCN, VERSION_ENDSCN, VERSION_STARTTIME, VERSION_ENDTIME, VERSION_OPERATION</pre>	<p>См. выше раздел «Технология Oracle Flashback»</p>
<p>Управление планами Oracle SQL Plan Management</p> <p>Хранение и обслуживание планов запросов для поддержки оптимизации запросов с целью принятия более качественных решений: при каждом выполнении запроса средство оптимизации запросов сравнивает текущий план запроса с сохраненным планом и автоматически выбирает лучший план.</p> <p>(Для целей DBA/настройки, не влияет на синтаксис запроса в приложении)</p>	<p>В некоторых случаях можно использовать ассоциацию с планом абстрактного запроса («загрузка абстрактного плана»), а также утилиту QPTune. Однако в ASE не поддерживается автоматическое сравнение вновь созданных планов с захваченными прошлыми планами.</p>

<p>AWR (Automatic Workload Repository – автоматический репозиторий рабочей нагрузки). В нем сохраняется каждый запрос вместе с соответствующими показателями рабочей нагрузки и метриками. Это позволяет автоматически выявлять запросы с самой низкой скоростью выполнения (для которых при последующем выполнении будут применяться планы запросов под управлением Oracle SQL Plan Management). (Используется для целей DBA/настройки, не влияет на синтаксис запроса в приложении)</p>	<p>Метрики запросов в ASE захватываются в sysquerymetrics путем использования функции «metrics capture» или с помощью таблиц MDA.</p> <p>В ASE не поддерживаются автоматические действия с захваченными данными.</p>
<p>Расширенные функции управления очередями Oracle Advanced Queuing</p>	<p>Аналогично существующей в ASE концепции Real-Time Messaging Service (RTMS – Служба управления сообщениями в режиме реального времени), которая позволяет напрямую подключаться к интерфейсу SQL с помощью продуктов, обеспечивающих передачу сообщений по шине, например, Tibco и MQ Series</p>
<p>Пакеты PL/SQL для доступа к Интернету OWA_CUSTOM, OWA_CX, OWA_OPT_LOCK, OWA_SEC, OWA_TEXT, OWA_UTIL</p>	<p>Эквивалент отсутствует. Необходима пользовательская реализация в ASE.</p>
<p>Формы Oracle</p>	<p>Перепишите, используя Sybase PowerBuilder. См. раздел 9.2.</p>

КОНТАКТНАЯ ИНФОРМАЦИЯ

Для запросов из стран Европы, Ближнего Востока и Африки:
+(31) 34 658 2999

Для запросов из стран Азиатско-Тихоокеанского региона:
+852 2506 8900 (Гонконг)

Для запросов из стран Латинской Америки:
+770 777 3131 (Атланта, шт. Джорджия)

Sybase, Inc.
Worldwide Headquarters
One Sybase Drive
Dublin, CA 94568-7902
U.S.A
1 800 8 sybase

Sybase CIS
115114, Москва,
Дербеневская набережная,
д. 7, стр. 16
+7 (495) 797-4774

Copyright © 2011 Sybase, Inc. Все права сохранены. Неопубликованные права защищены согласно законодательству США об авторских правах. Название Sybase и логотип Sybase являются товарными знаками компании Sybase, Inc. или ее дочерних компаний. Все другие товарные знаки являются собственностью соответствующих владельцев. Знак ® означает регистрацию в Соединенных Штатах Америки. Спецификации могут быть изменены без уведомления.

