


Управление распределенными базами данных в СУБД Oracle



Главный недостаток капитализма – неравное распределение благ; главное преимущество социализма – равное распределение лишений.

У. Черчилль

Корпоративные системы баз данных © М.Л. Цымблер

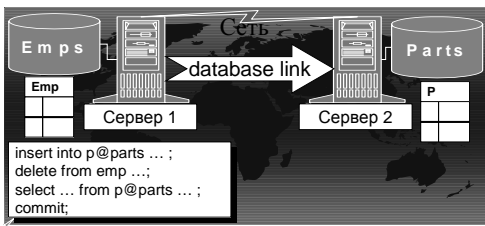
Содержание

- Понятие распределенной базы данных
- Управление распределенными базами данных
 - Драйверы Net8
 - Глобальные имена объектов
 - Связи базы данных
 - Удаленные и распределенные запросы
 - Вызов удаленных процедур
 - Двухфазная фиксация распределенных транзакций
 - Администрирование распределенной базы данных
 - Репликация

Корпоративные системы баз данных © М.Л. Цымблер 2

Распределенные базы данных

- *Распределенная база данных (distributed database)* – совокупность взаимосвязанных баз данных, распределенных в компьютерной сети.
- Компьютеры сети – узлы. Узел может выполнять функции клиента или/и сервера.



```
insert into p@parts ... ;
delete from emp ... ;
select ... from p@parts ... ;
commit;
```

Корпоративные системы баз данных © М.Л. Цымблер 3

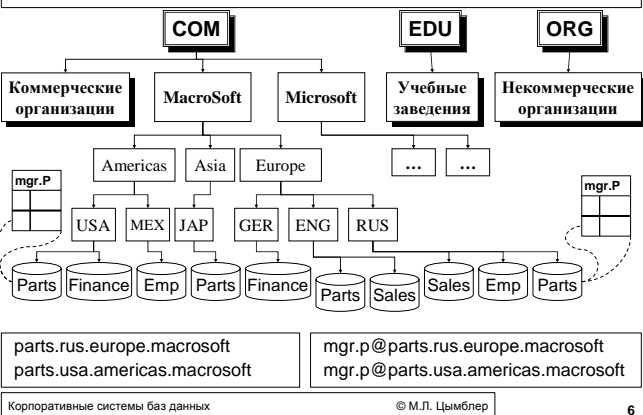
Семейство драйверов Net8

- Семейство драйверов Net8 – компонент СУБД Oracle, обеспечивающий соединения клиентов и сервера базы данных внутри сети и межсетевые соединения узлов распределенной базы данных.
- Net8 поддерживает гетерогенные сети и гетерогенные распределенные базы данных.

Глобальные имена

- Каждая база данных (объект схемы) распределенной базы данных имеет глобальное имя, которое состоит из имени сетевого домена и имени базы данных (имени объекта).
- В распределенной базе данных могут быть базы данных (объекты схемы) с одинаковым именем, но их глобальные имена всегда различны.

Пример: глобальные имена



Связи базы данных

- *Связь базы данных (database link)* – объект схемы, описывающий путь к базе данных на удаленном узле распределенной базы данных.

```
-- Создание связи базы данных
create database link parts.usa.americas.microsoft
  connect to mgr
  identified by thisispassword;
```

```
-- Использование связи базы данных
select p#, name from mgr.p@parts.usa.americas.microsoft
where price>100;
```

Корпоративные системы баз данных

© М.Л. Цымблер

7

Удаленные vs распределенные запросы

- *Удаленный (remote) запрос* – запрос к данным, расположенным на одном и том же удаленном узле.
- *Распределенный (distribute) запрос* – запрос к данным, расположенным на двух или более удаленных узлах.

```
-- Удаленные запросы
select p#, name from mgr.p@parts.usa.americas.microsoft;
update mgr.p@parts.usa.americas.microsoft
  set price=price+0.05;
-- Распределенные запросы
select p#, name
from mgr.sp Local_sp, mgr.p@parts.usa.americas.microsoft Remote_p
where Local_sp.p#=Remote_p.p#;
update mgr.p@parts.usa.americas.microsoft Remote_p
  set Remote_p.price=0.05+(select price from mgr.p Local_p
  where Remote_p.p#= Local_p.p#);
```

Корпоративные системы баз данных

© М.Л. Цымблер

8

Вызов удаленных процедур

- В приложении распределенной базы данных можно использовать вызовы как локальных, так и *удаленных процедур (remote procedure calls, RPC)*.
- При вызове удаленной процедуры локальный сервер передает фактические параметры процедуры удаленному серверу.

```
begin
  mgr.calc_discounts@parts.usa.americas.microsoft(sysdate);
end;
```

Корпоративные системы баз данных

© М.Л. Цымблер

9

Удаленные vs распределенные транзакции

- *Удаленная транзакция* – транзакция, содержащая один или более удаленный запрос к одному и тому же удаленному узлу.
- *Распределенная транзакция* – транзакция, содержащая один или более распределенный запрос.

```
-- Удаленная транзакция
select p#, name
from mgr.p@parts.usa.americas.microsoft;
update mgr.p@parts.usa.americas.microsoft
set price=price+0.05;
commit;
```

```
-- Распределенная транзакция
select p#, name
from mgr.p@parts.usa.americas.microsoft;
update mgr.p
set price=price+0.05;
commit;
```

Двухфазная фиксация

- *Двухфазная фиксация (two-phase commit, 2PC)* – механизм управления распределенными транзакциями, гарантирующий, что на всех узлах, вовлеченных в транзакцию, все произведенные ею изменения будут либо зафиксированы, либо подвергнуты откату (независимо от возможных сбоев на любом из узлов).
- Фазы:
 1. Подготовка
 2. Фиксация

Обработка сбоев 2PC

- Таблица DBA_2PC_PENDING словаря данных хранит информацию о текущих распределенных транзакциях.
- Фоновый процесс RECO использует информацию данной таблицы для разрешения *висячих (in-doubt)* транзакций.
- АБД может зафиксировать или откатить висячую транзакцию *вручную*.

```
select local_tran_id, comment, state, fail_time, advice
from dba_2pc_pending;
```

```
rollback force '1.2.3';
commit force '4.7.9';
```

Администрирование распределенной базы данных

- Принцип *автономности узлов*: каждый сервер распределенной базы данных администрируется отдельно и независимо от других.
- Основные преимущества автономности узлов:
 - Узлы отражают структуру организации
 - Область ответственности АБД каждого узла локальна
 - Сбой на одном из узлов не требует остановки всех распределенных транзакций
 - Восстановление данных после сбоя требуется только для отказавшего узла
 - Словарь базы данных имеется на каждом узле
 - Узлы могут независимо переходить на другое аппаратное/программное обеспечение

Безопасность распределенной базы данных

- Каждый узел распределенной базы данных должен обеспечивать:
 - специальную учетную запись и соответствующие связи базы данных для подключения к другим узлам
 - роли для наделения пользователя приложений распределенной базы данных соответствующими привилегиями.
- Семейство драйверов Net8 обеспечивает шифрование и безопасную передачу данных между узлами.

Целостность распределенной базы данных

- Целостность данных на отдельном узле – ограничения целостности, триггеры.
- Целостность данных между узлами – только триггеры.

Прозрачный доступ к данным: синонимы

- *Синонимы* могут быть созданы для объектов схем с удаленных узлов.

```
-- Создание синонима
create synonym usa_parts for
mgr.p@parts.usa.americas.microsoft;
create synonym eng_parts for
mgr.p@parts.eng.europe.microsoft;
create synonym rus_parts for
mgr.p@parts.rus.europe.microsoft;
-- Использование синонима
select p#, name
from usa_parts up, rus_parts rp
where rp.p#=up.p# and rp.price<up.price;
```

Прозрачный доступ к данным: представления

- Базовые таблицы *представлений* могут располагаться на удаленных узлах.

```
-- Создание представления (используются синонимы)
create view europe_parts as
select p#, name, city, price
from eng_parts, rus_parts
order by price desc;
-- Использование синонима
select * from europe_parts
where price>100;
```

Репликация

- *Репликация (replication)* или *тиражирование* – хранение идентичных копий базы данных на нескольких серверах одновременно.
- Репликация не требует перманентной идентичности копий (только в определенные АБД моменты времени).
- Репликация обеспечивает быстрый доступ к данным (разделяемые данные – на локальном узле) и высокую готовность данных (в случае сбоя – взять данные на удаленном узле).

Реализация репликации

- Некоторый узел выделяется в качестве главного. База данных главного узла реплицируется на удаленные узлы.
- Методы репликации:
 - *Моментальные снимки* – измененные части всех или выбранных таблиц в дискретные моменты времени распространяются на удаленные узлы.
 - *Запомни и передай* – изменения передаются при завершении транзакции на главном узле как очереди сообщений.
 - *Прокрутка вперед* – в дискретные моменты времени журнал главного узла прокручивается вперед на удаленных узлах.

Корпоративные системы баз данных

© М.Л. Цымблер

19

Пример репликации

```
create snapshot all_parts
refresh fast -- быстрая (не complete – полная) репликация
start with sysdate
next sysdate+7
as
select * from eng_parts
union
select * from rus_parts
union
select * from usa_parts
union
select * from jap_parts;
```

Корпоративные системы баз данных

© М.Л. Цымблер

20
