Новые возможности для разработчиков в Oracle 10G, часть 1

ORAMASTER

Новые возможности для разработчиков в Oracle 10G

- SQL
- PL/SQL
- Расширение функционала
- Производительность
- Безопасность
- Новое в SQL*Plus

ORAMASTER

Новые возможности в SQL

- Регулярные выражения
- Расширения оператора MERGE
- Числовые типы данных с плавающей точкой
- Запросы и сортировка без учёта регистра
- Оператор “Q" – оператор кавычек

ORAMASTER

Новые возможности в SQL (продолжение)

- Новое в иерархических запросах
- Регистрация ошибок DML (dml error logging)
- Секционированное внешнее соединение (partition outer join)
- Асинхронный commit
- Функция COLLECT
- ORA_ROWSCN

Новые возможности SQL

Расширения оператора MERGE

- Возможность пропустить секцию update или insert
- Возможность указания условий для update и insert

```
…when matched then
update set s.status = h.status
 where pay_code='cash'
```

- Новая возможность в конструкции ON
  - on (1=0)
- Конструкция delete (допустима в секции update)

```
…when matched then
update set s.status = h.status
 where pay_code = 'cash'
delete where (s.pay_date>sysdate-60)
```

ORAMASTER

Числовые типы данных с плавающей точкой

- Название «плавающая точка (запятая)» происходит от того, что точка в позиционном представлении числа может быть помещена где угодно относительно цифр в строке
- Основаны на стандарте IEEE 754

NUMBER(p,s) :
38 знач.цифр
10^{-130} до 10^{126}
от 0 до 22 байт

- **BINARY_FLOAT – числовой тип одинарной точности с плавающей точкой, 7 значащих цифр, точность не задаётся, диапазон $\approx\pm10^{38.53}$, занимает 5 байт (1 длина+4 значение)**
- **BINARY_DOUBLE – числовой тип двойной точности с плавающей точкой, 16 значащих цифр, точность не задаётся, диапазон $\approx\pm10^{308.25}$, занимает 9 байт**
- Арифметические операции реализованы на аппаратном уровне – увеличение быстродействия
- Больший диапазон значений при неизменной относительной точности.

ORAMASTER

Числовые типы данных с плавающей точкой

- При задании литералом добавляется суффикс f или d

```
declare bf binary_float; bd binary_double;
begin
  bf := 35.22f;
  bd := 35.22d;
end;
```

- Добавлены функции преобразования: to_binary_float и to_binary_double
- Хорошее описание этих типов в книгах:
  - Tom Kyte - Expert Oracle Database Architecture: 9*i and 10g Programming* Techniques and Solutions
  - Steven Feuerstein - Oracle PL/SQL Programming, 5-th Edition

ORAMASTER

Числовые типы с плавающей точкой - особенности

- Числа с плавающей точкой – сохраняют приблизительное значение числа
- В ряде случаев занимают меньше места чем NUMBER
- Округление двоичное – не десятичное – не подходит для финансовых приложений
- Поддерживаются спец. значения: INF, -INF, NaN
- При делении на 0 не возникает исключения
- Требуется повышенная аккуратность и внимание при приведении типов.

Запросы и сортировка без учёта регистра

- Традиционно - upper/lower и fbi
- Сортировка без учёта регистра:
  - Case insensitive
    - `NLS_SORT=<sort_name>_CI`
  - Accent insensitive
    - `NLS_SORT=<sort_name>_AI`
- Сравнение в PL/SQL и в запросах
  - `nls_comp = binary`
  - `nls_comp = linguistic`
  - `nls_comp = ansi` (не все операции используют лингвистическое сравнение. Значение параметра оставлено для совместимости)

Nls_sort	Сравнение для рус. букв	Результат
binary	а=А и е=ё	нет
binary_ci	а=А	да
binary_ci	е=ё	нет
binary_ai	а=А и е=ё	да
russian	а=А	нет
russian_ci	а=А	да
russian_ci	е=ё	нет
russian_ai	а=А	да
russian_ai	е=ё	**нет**
generic_m_ai	а=А и е=ё	да

ORAMASTER

Запросы и сортировка без учёта регистра

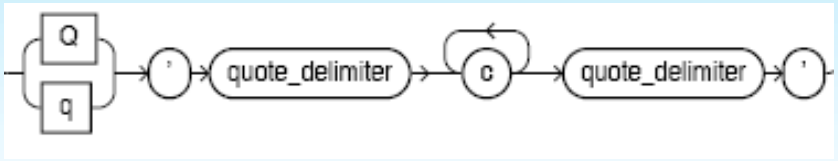
- Установка NLS параметров:
  - только когда нужно – `alter session set` …
  - на клиенте через переменную окружения/реестр
  - logon-триггер
- Индексы
  - При `nls_comp=linguistic` обычные индексы по символьным полям почти перестают использоваться, необходимо создавать лингвистические индексы.

```
create index emp_fname_idx_binai on employees
(nlssort(first_name,'nls_sort=binary_ai'));
```

  - Оператор `Like 'abc%'` индекс не использует  (но в 11G исправлено !)

Оператор «Q" –оператор кавычек

- Позволяет избавиться от тройных, четверных кавычек при указании литералов
- “Quote_delimiter” – любой символ, кроме space, tab, return



- А также пары [] {}()<>

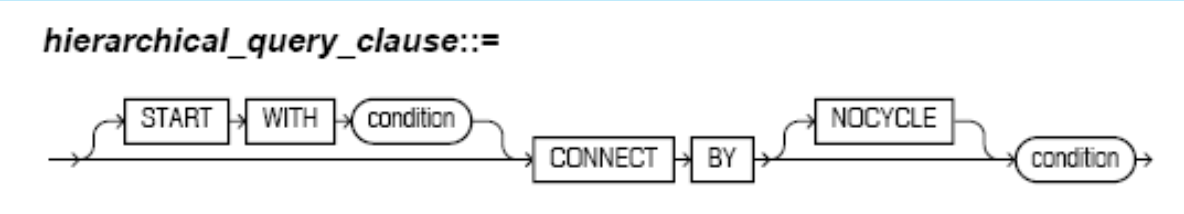
```
execute immediate
   'select count(*) from hr.employees where
job_id =''SH_CLERK''' and last_name like ''B%'''
```

```
execute immediate
   Q'[select count(*) from hr.employees where
job_id ='SH_CLERK' and last_name like 'B%' ]'
```

ORAMASTER

Новое в иерархических запросах

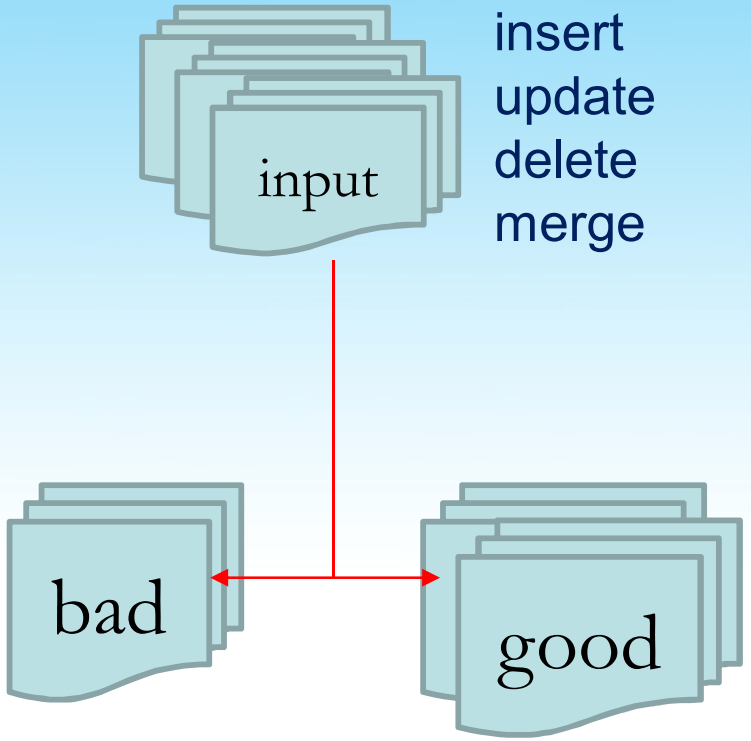
- Дополнение в синтаксисе



- Оператор CONNECT_BY_ROOT
- Псевдостолбец CONNECT_BY_ISLEAF
- Псевдостолбец CONNECT_BY_ISCYCLE
- В конструкции connect by допустим некоррелируемый подзапрос

Регистрация ошибок DML (dml error logging)

- Возможность продолжить операцию DML при возникновении ошибки и зарегистрировать ошибочные записи.
- Запись ошибок в журнал - таблицу регистрации ошибок DML
- Наподобие bad-файла в sqlldr
- Не выполняется откат операции DML
- Регистрируются ошибки нарушения ограничений, длины столбцов, при выполнении триггеров и другие



Регистрация ошибок DML

- Имя таблицы - err$_первые_25_символов_таблицы
- Создание журнала ошибок для таблицы вручную или с помощью:
  - `dbms_errlog.create_error_log (<DML_table_name>,[<error_table_name>] ...)`

```
exec dbms_errlog.create_error_log('emp')
```

```
err$_emp
```

- В таблице err$ имеются столбцы :
  - обязательные(первые пять): информация об ошибке

Столбец	Тип данных	Описание
ORA_ERR_NUMBER$	NUMBER	Номер ошибки Oracle
ORA_ERR_MESG$	VARCHAR2(2000)	Текст ошибки Oracle
ORA_ERR_ROWID$	ROWID	Rowid строки (только для update и delete)
ORA_ERR_OPTYP$	VARCHAR2(2)	Тип операции I=insert, U=update, D=delete
ORA_ERR_TAG$	VARCHAR2(2000)	Пользовательская метка (тэг)

  - необязательные: данные из сбойной строки

Регистрация ошибок DML (продолжение)

- Синтаксис:

```
LOG ERRORS [INTO <error_table>] [('<tag>')]
[REJECT LIMIT <limit>]
```

- Можно задать пометку(тэг)
- Фраза REJECT LIMIT - по сути необходима
  - количество ошибок, после чего оператор прекращается и откатывается
  - по умолчанию равен 0
  - можно указать UNLIMITED
  - если оператор откатывается, записи в err$table сохраняются !!!

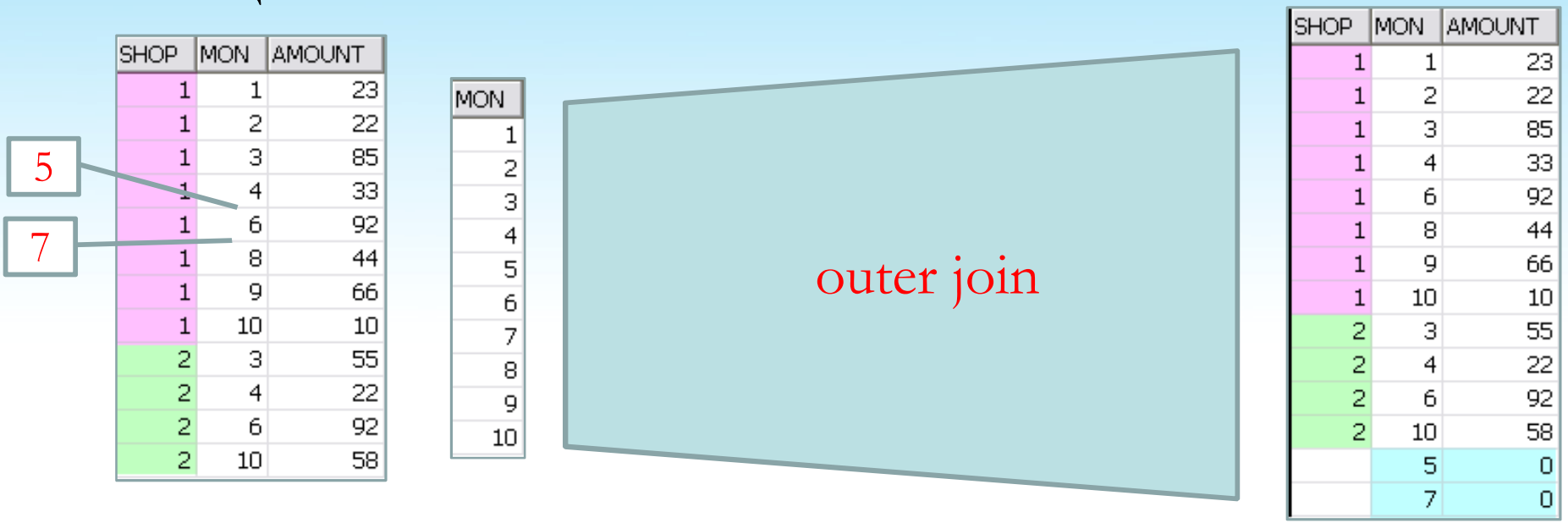
Регистрация ошибок DML (продолжение)

- Имеются ограничения – оператор сбойнёт и регистрации не будет, если:
  - Нарушается отложенное ограничение
  - Нарушается ограничение уникальности или уникальный индекс:
    - Для direct path INSERT или MERGE
    - Для UPDATE или MERGE

Для direct path INSERT всё работает

Секционированное внешнее соединение (partition outer join)

- Проблема - не по всем месяцам есть данные
- Нужно показать отчёт по каждому магазину по всем месяцам



SHOP	MON	AMOUNT
1	1	23
1	2	22
1	3	85
1	4	33
1	6	92
1	8	44
1	9	66
1	10	10
2	3	55
2	4	22
2	6	92
2	10	58

MON
1
2
3
4
5
6
7
8
9
10

SHOP	MON	AMOUNT
1	1	23
1	2	22
1	3	85
1	4	33
1	6	92
1	8	44
1	9	66
1	10	10
2	3	55
2	4	22
2	6	92
2	10	58
	5	0
	7	0

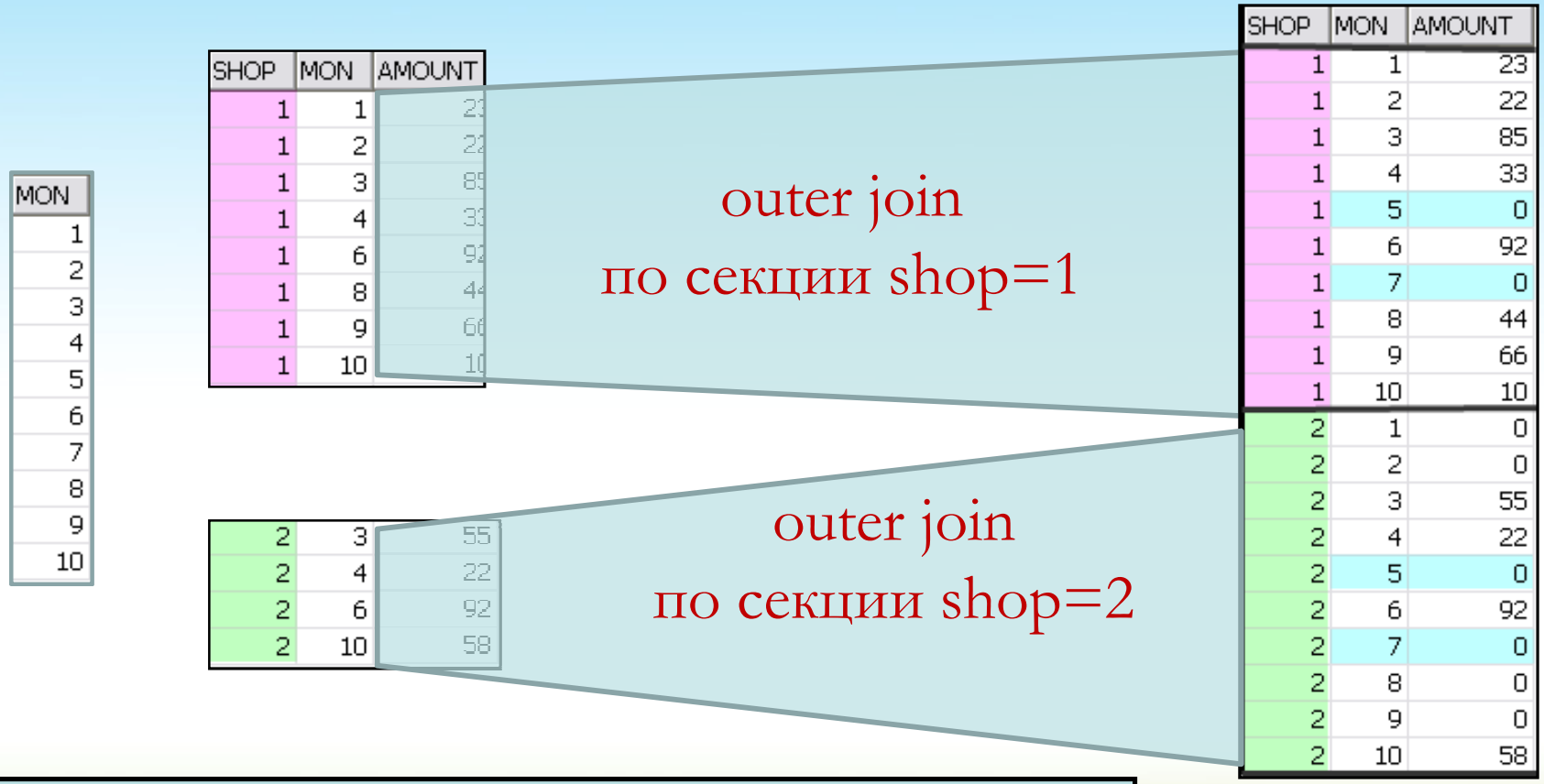
- Обычный outer join этого не позволяет

```
select shop, m.mon, nvl(sum(amount),0)
  from sale_hist s
right outer join months m on (s.mon=m.mon)
```


ORAMASTER

Секционированное внешнее соединение (partition outer join)

- To fill gaps in sparse data - заполнение «дыр» в разрежённых данных -
- Лучший термин - внешнее соединение по секциям
- Результат внешнего соединения по секциям - это UNION ALL внешнего соединения каждой секции таблицы(запроса), разбитой на логические секции, с таблицей на другой стороне соединения




```
select shop, m.mon, nvl(sum(amount),0)
  from sale_hist s partition by (shop)
right outer join months m on (s.mon=m.mon)
```


ORAMASTER

Внешнее соединение по секциям

- To fill gaps in sparse data - заполнение пустых мест на прерывистом интервале
- Разбиение таблицы SALES на N секций
- Каждую из этих секций внешне соединяем с данными MONTHS
- Соединение выполняется с данными без пропусков (dense dataset) - в MONTHS данные - сплошные

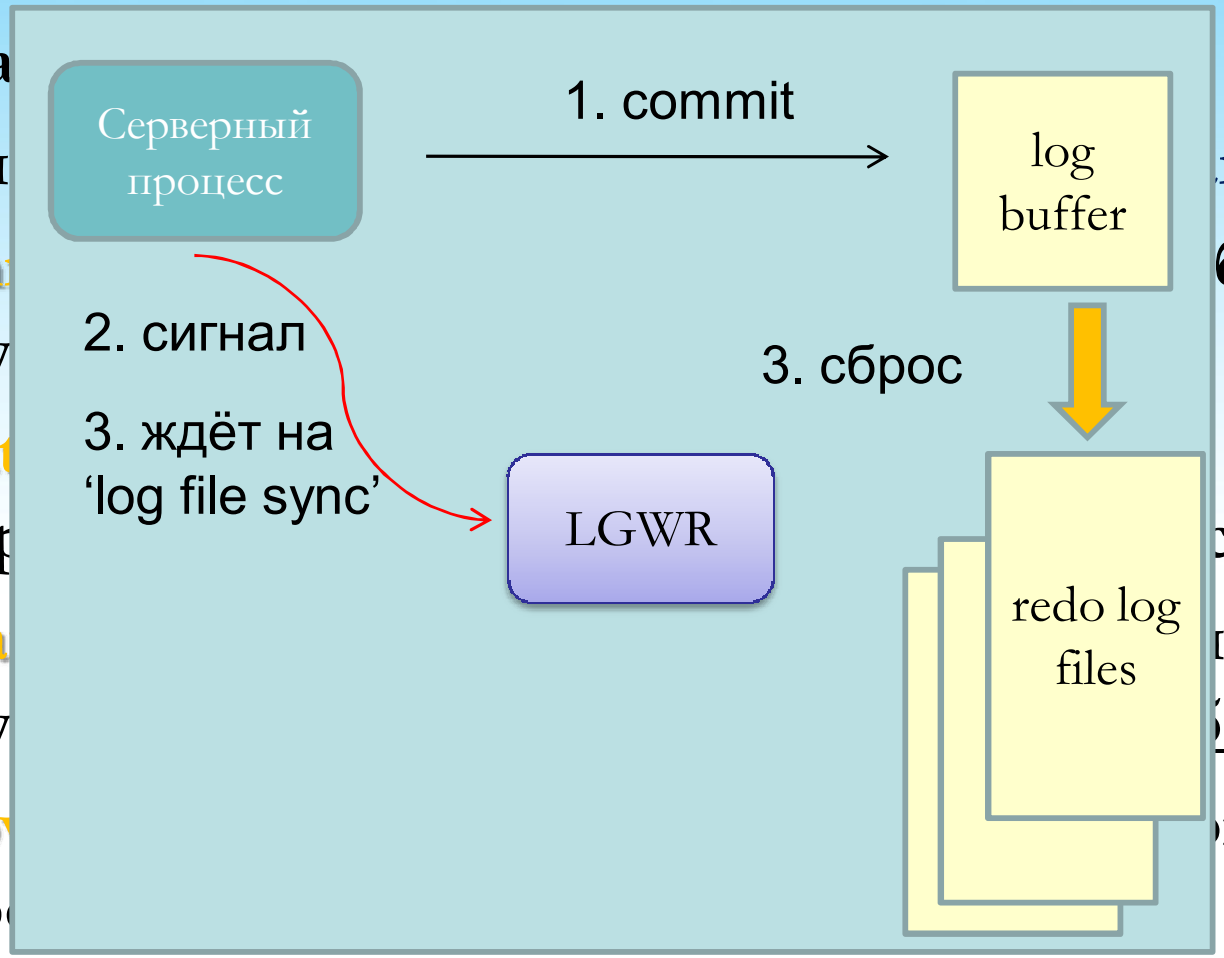

```
select s.shop, m.mon, sum(amount)
  from months m
       left outer join
   sale_hist s partition by (shop)
       on(s.mon=d.mon)
```

Асинхронный commit

- Устраняет события ожидания log file sync
- Опции commit: immediate | batch wait | nowait
  - immediate - передавать LGWR сигнал о сбросе redo в журнальные файлы
  - batch - redo продолжает накапливаться в log-буфере, серверный процесс не инициирует сброс redo
  - wait - приложение ожидает пока redo запишется в журнальные файлы и не продолжает работу
  - nowait- приложение передает команду commit и продолжает работу
- Не предназначен для OLTP приложений !!!

Асинхронный commit

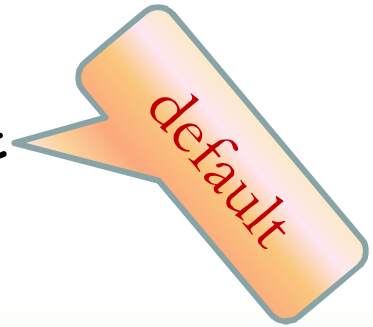
- Устра
- Опци
  - im
    жу
  - bat
    сер
  - wa
    жу
  - no
    пр



- Не предназначен для OLTP приложений !!!

Асинхронный commit

- Применим например:
  - в системах массовой загрузки, при возможности повторить загрузку в случае сбоя
  - при съёме данных с датчиков, при котором потеря некоторого количества данных несущественна, но важно получить текущие данные быстрее
- Можно задать на уровне команды, сессии и системы
- `commit work write batch nowait`
- `alter system set commit_write=immediate,wait`
- `alter system set commit_write=batch,nowait`
- `alter session …`



ORAMASTER

Асинхронный commit

- Статистика загрузки с SQL Loader

	Immediate, wait 64 rows		Immediate, wait 500 rows		Batch, nowait	
	Elapsed	CPU	Elapsed	CPU	Elapsed	CPU
1-й сеанс	20.91	1.24	16.54	1.23	6.72	0.90
2-й сеанс	20.73	1.11	16.92	1.36	6.87	0.96
3-й сеанс	20.24	1.19	16.10	1.01	6.53	1.12
4-й сеанс	20.83	1.42	16.08	1.26	7.31	0.82
log file sync	62 сек		44 сек		1.7 сек	

ORA_ROWSCN

- Псевдостолбец, с его помощью можно определить SCN последнего изменения строк
- Определение в SQL reference
- По умолчанию отслеживается на уровне блока
- Если надо на уровне строки, то таблица создаётся с атрибутом – ROWDEPENDENCIES
  - Требуется дополнительно 6 байт для каждой строки для хранения SCN
- scn_to_timestamp(ora_rowscn)
- На базе ORA_ROWSCN можно организовать алгоритм оптимистического блокирования

Функция LNNVL

- Предоставляет краткий способ для вычисления выражения (условия), когда один или оба операнда в условии могут быть null.
- По сути, представляет вариант оператора NOT
- В качестве аргумента передаётся условие
  - `where lnnvl(x=y)`
- Возвращает TRUE , если условие FALSE или UNKNOWN
- Возвращает FALSE, если условие TRUE
- Функция может использоваться только в конструкции where

Новые возможности SQL*PLUS

- Easy Connect Naming Method
  - `connect hr/hr@host:1521/orcl.oracle.com`
- Новые переменные
  - `_date  _user  _privilege`
    - `set sqlprompt "&_user@&_connect_identifier> "`
- Login-script
  - Выполняется при каждом соединении с БД
- Новое в команде spool
  - ключ append
- Имена файлов в командах могут содержать пробелы

Новые возможности SQL*PLUS

- Set serveroutput
  - Длина строки 32K
  - Размер буфера unlimited (по умолчанию)
- Команды:
  - show recyclebin
  - purge recyclebin
- Запуск в режиме as sysdba без кавычек
  - `c:\sqlplus / as sysdba`
- Запуск в режиме совместимости (ключ –c)
  - `c:\sqlplus -c 9.2 "/ as sysdba"`
- Оператор xquery

The End

www.oramaster.ru