

ORACLE®

ORACLE®

# Новые возможности Oracle Database 12c

Игорь Мельников  
Oracle CIS

Hardware and Software  
Engineered to Work Together

ORACLE  
OPEN  
WORLD

# План

- Введение
- Консолидация баз данных
- Автоматизация ILM
- Oracle Database In-Memory Option
- Заключение

# Проблема #1

Консолидация баз  
данных (Pluggable  
Database)



ORACLE

# Консолидация баз данных в облаке

## Ключевые требования...



Приложения не переписываются

Изоляция и multitenancy БД

Быстрое развертывание и клонирование

Защита и высокая доступность

Снижение затрат на сопровождение

Управление многими БД как одной

Лучшее использование ресурсов

Производительность и масштабируемость

# Архитектура СУБД Oracle Database

Состоит из памяти, процессов и файлов БД

ERP

Память



Фоновые  
процессы

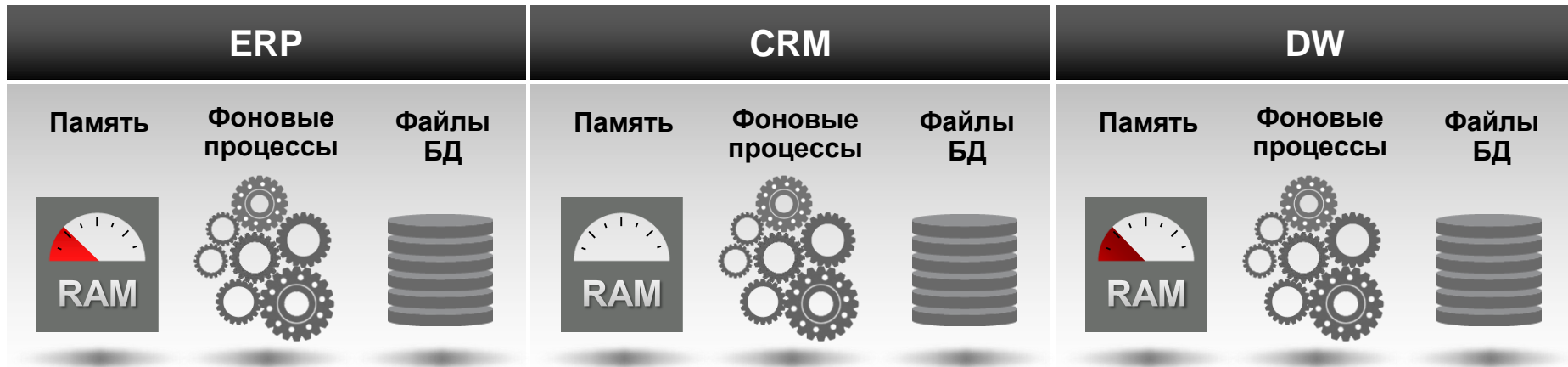


Файлы  
БД



# Архитектура СУБД Oracle Database

Для каждой БД требуется отдельная память и фоновые процессы



# Новая архитектура СУБД

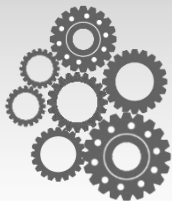
Память и процессы общие для всех БД в контейнере

## ERP

Память



Фоновые  
процессы



Файлы  
БД

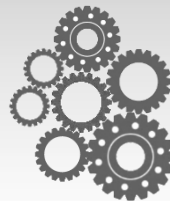


## CRM

Память



Фоновые  
процессы



Файлы  
БД

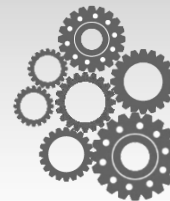


## DW

Память



Фоновые  
процессы



Файлы  
БД





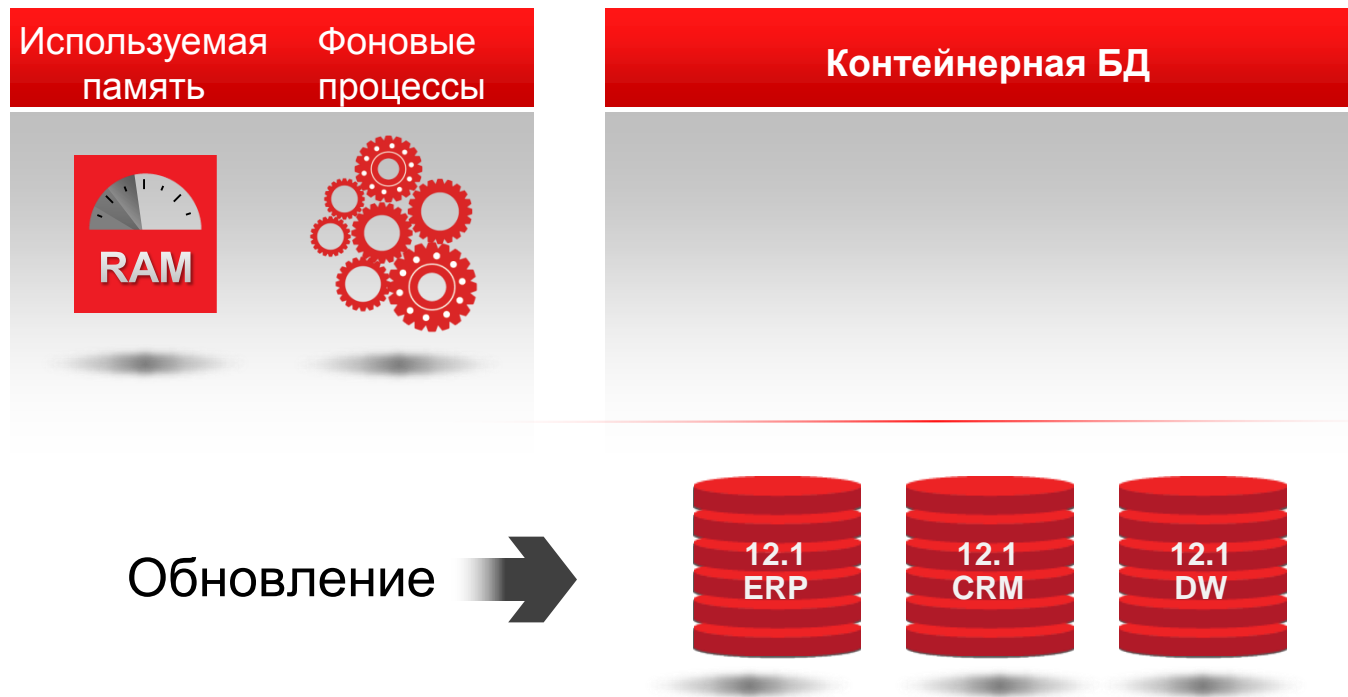
# Новая архитектура СУБД

Память и процессы общие для всех БД в контейнере



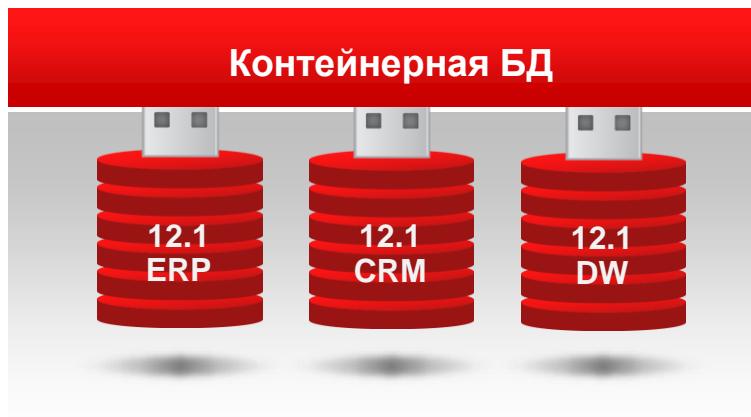
# Консолидация БД старых версий

Шаг 1: Обновление существующих БД на текущем месте



# Консолидация БД старых версий

Шаг 2: Вставляем обновленные базы данных в контейнер



# Отключить / подключить БД

## Пример

### Отключить

```
alter pluggable database HCM  
unplug into '/u01/app/oracle/oradata/.../hcm.xml'
```

### Подключить

```
create pluggable database My_PDB  
using '/u01/app/oracle/oradata/.../My_PDB.xml'
```

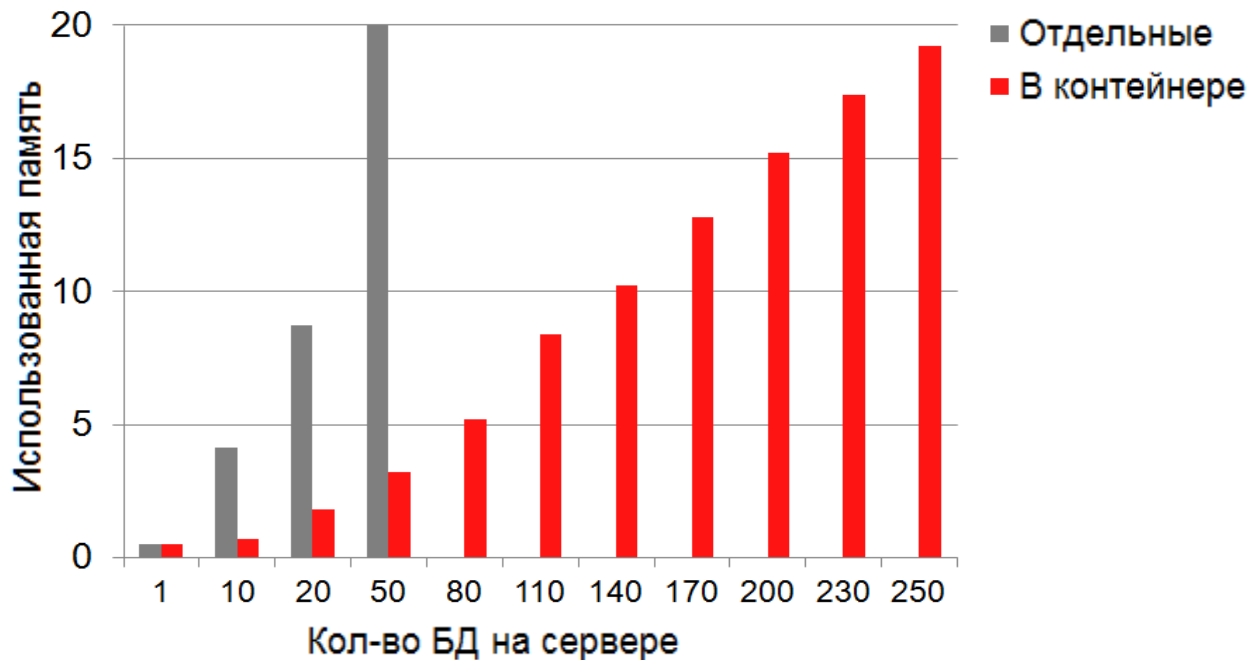
# Сравнение: БД в контейнере и отдельные БД

Очень эффективно: в 6х меньше ресурсов H/W, в 5х раз больше масштабируемость

Сравнение на OLTP-тесте

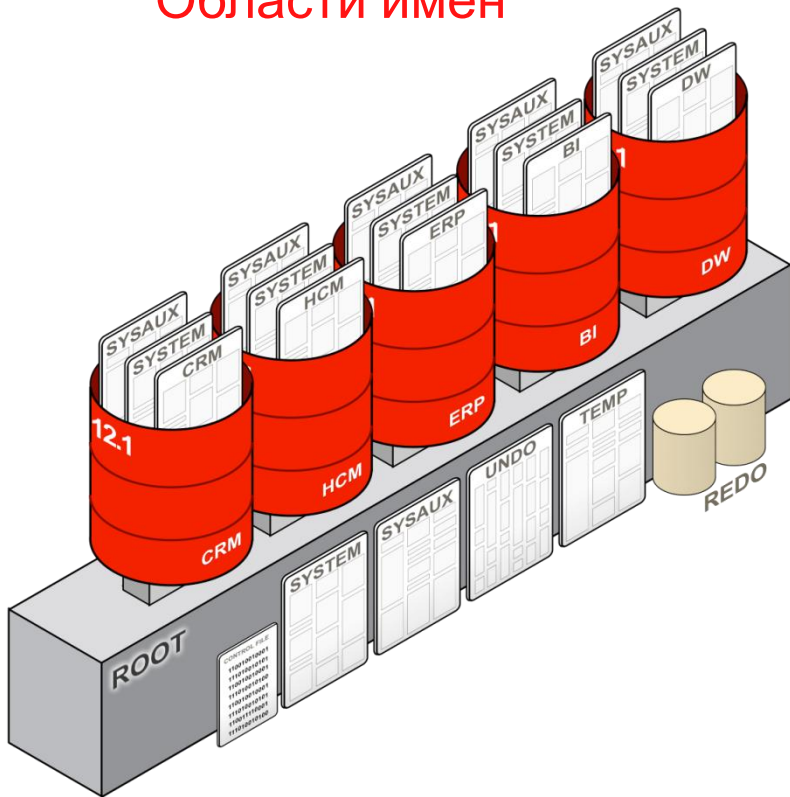
Всего 3Гб памяти из 20Гб было использовано для 50 баз данных

Контейнер БД-х был масштабирован до 250 баз, тогда как отдельные БД – максимум до 50 !



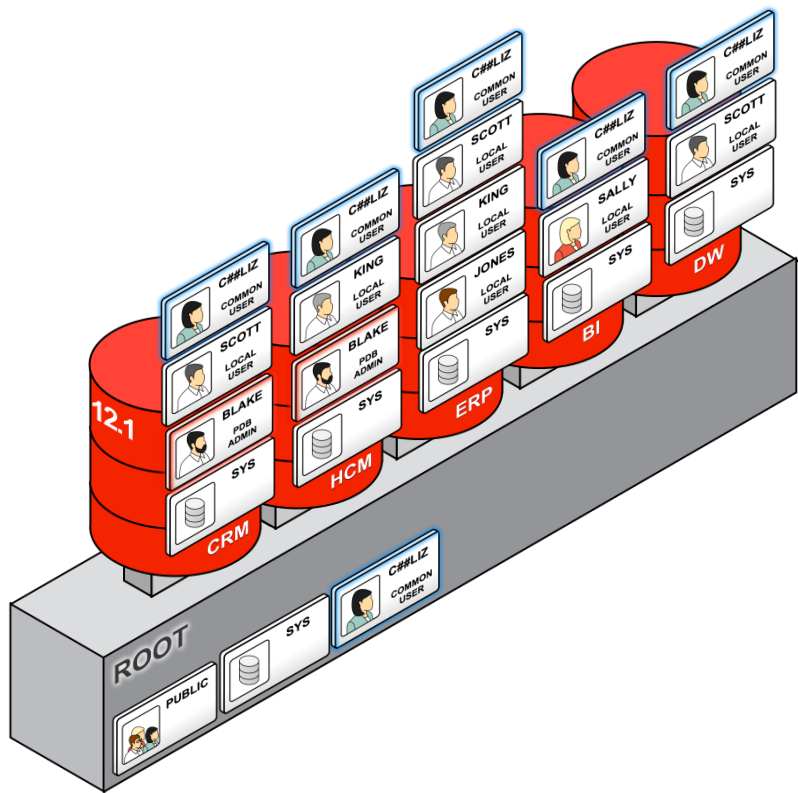
# Файлы в CDB (контейнерной БД)

## Области имён



- Каждая PDB имеет свой набор табличных пространств, включая SYSTEM и SYSAUX
- PDB базы данных разделяют UNDO, REDO и контрольные файлы, spfile - файл параметров
- По умолчанию CDB содержит только одно TEMP табличное пространство, но PDB базы данных могут создавать свои собственные

# Пользователи



- Локальные пользователи являются преемниками пользователей, созданных в обычной не-CDB базе данных
- Локальный пользователь может быть создан только в PDB
- Локальный пользователь может администрировать только свою PDB
- Общий пользователь создаётся в root части CDB и представлен в каждой PDB
- Общий пользователь может входить в любую PDB, где у него есть привилегия “Create Session”, и может администрировать PDB
- Общие пользователи DBA могут администрировать всю CDB базу данных

# Управление разделяемыми ресурсами

## Распределение ресурсов для баз данных в контейнере





# Упрощение установки обновлений ПО

Патч применяется один раз, - все БД в контейнере обновлены !

Единое  
обновление



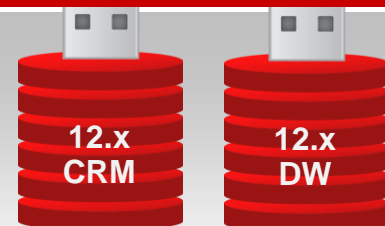
# Обновление через перенос в другой контейнер

Гибкое обновление версии БД

Исходная контейнерная БД (12.1)



Целевая контейнерная БД (12.x)



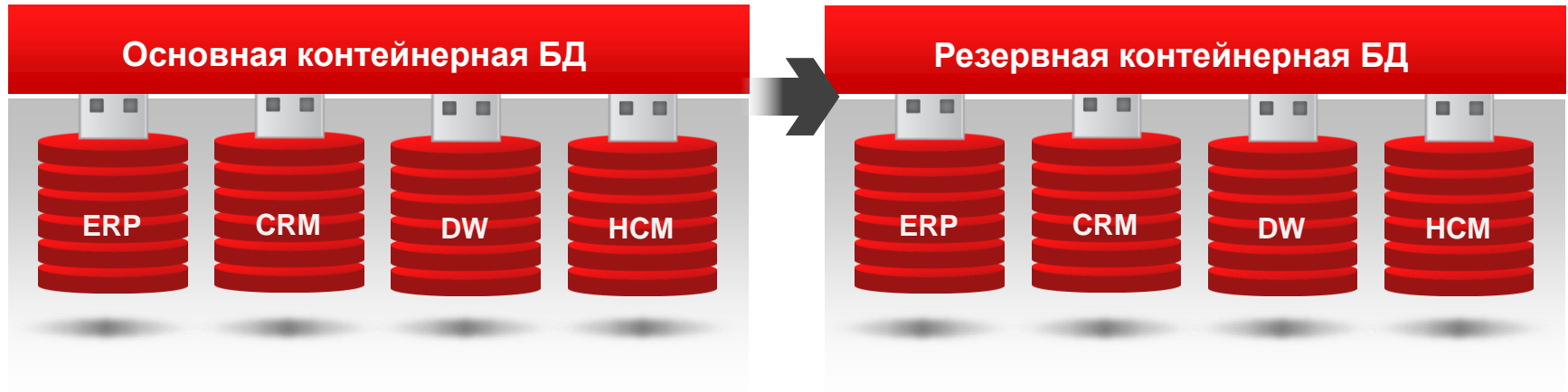
# Единое управление backup/restore

Резервирование всего контейнера; восстановление требуемой БД



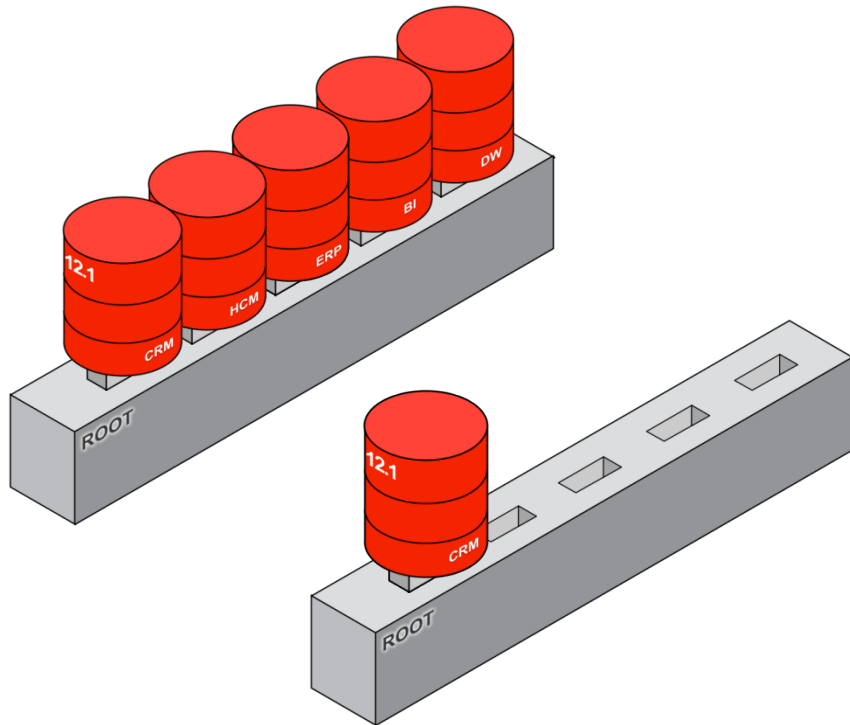
# Dataguard на уровне контейнеров

Один контейнер для всех резервных (standby) БД



# Клонирование БД для тестов и разработки

## Быстрое и гибкое копирование баз между контейнерами



- PDB база данных может быть клонирована внутри той же CDB
- PDB может быть клонирована из удалённой CDB

# Клонирование PDB

## Пример

Локально

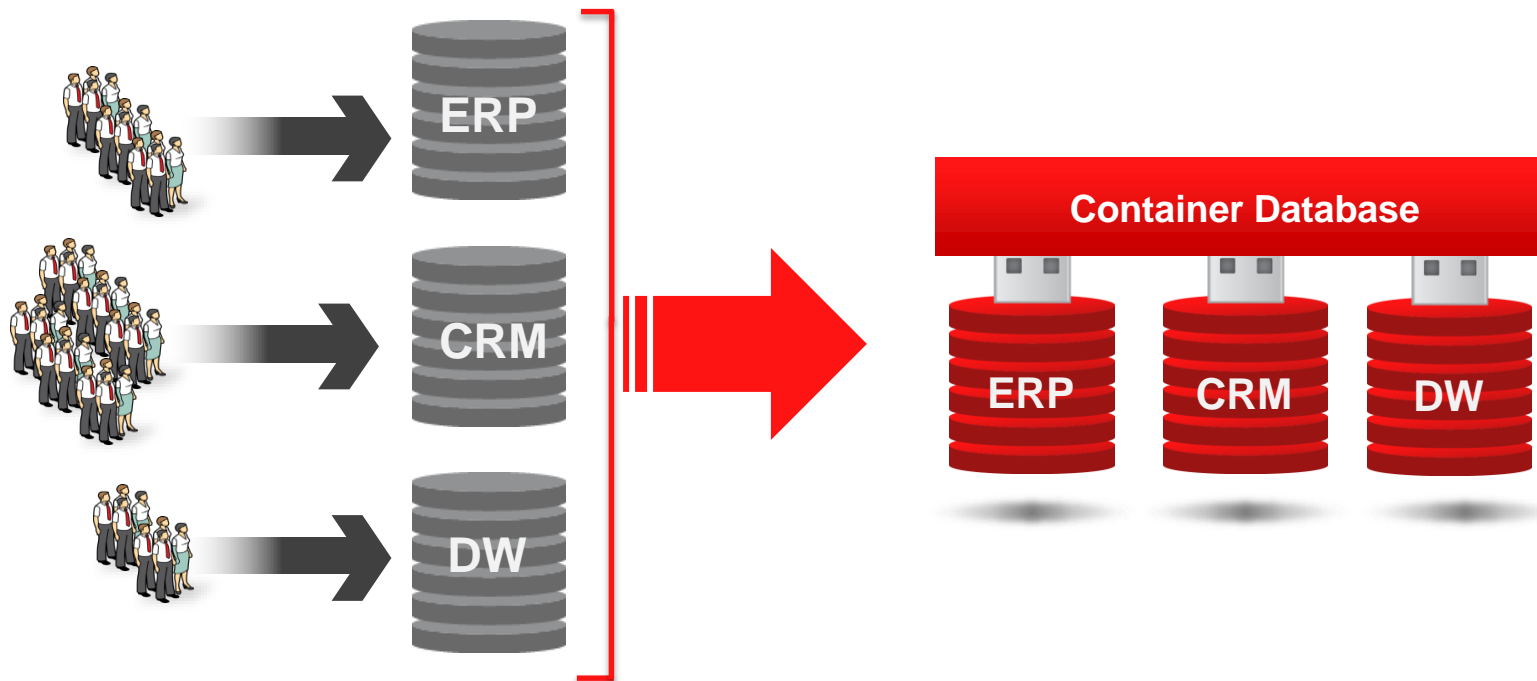
```
create pluggable database HCMBI from HCM
```

Удалённо (через DB Link)

```
create pluggable database HCMBI from HCM@us.acme.db1
```

# Тестирование нагрузки после консолидации

## Real Application Testing



# Контейнер БД идеально подходит для SaaS

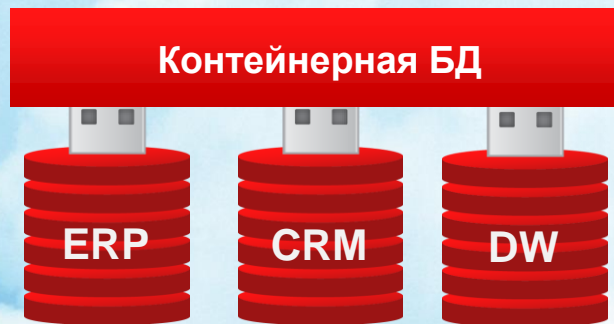
БД клиентов изолированы друг от друга





# Контейнерная СУБД Oracle Databases

Новая архитектура для консолидации баз данных в облаке



Приложения не переписываются

Изоляция и multitenancy БД

Быстрое развертывание и клонирование

Защита и высокая доступность

Снижение затрат на сопровождение

Управление многими БД как одной

Лучшее использование ресурсов

Производительность и масштабируемость

ORACLE

# Проблема #2

Снижение затрат на хранение данных (Automatic Data Optimization)



ORACLE

# Задачи по снижению затрат на storage

Хранить больше  
данных без  
увеличения  
затрат

Сжимать данные  
без потери  
производитель-  
ности

Иметь уровни  
хранения и  
сжимать в  
зависимости от  
частоты  
использования  
данных

# Automatic Data Optimization 12c



По мере “остывания” данных, Automatic Data Optimization автоматически перемещает данные и может их сжимать на лету

# Сжатие данных

Уменьшить объем данных, считывать сжатые данные быстрее

**“Горячие”  
данные**

10101010111010100110101  
11000010100010110111010  
10100101001001000010001  
01010110100101101001110  
00010100100101000010010  
00010001010101110011010  
110001010010011  
111001001000010  
001010101101000

**3X**

Advanced Row Compression

**“Теплые”  
данные**

10101010111010100110101110000101000101  
101110101010010100100100001000101011  
01001011010011100001010010010100001001  
00001000101010111001101110011000111010

100001010010010010  
100001001000010  
001010101101001

**10X**

Columnar Query Compression

**Архивные  
данные**

10101010111010100110101110000101000101101110101  
010010100100100001000101010101000101100111000  
010100100101000010010000100010101110011011100

100001010010010010  
100001001000010  
001010101101001

**15X**

Columnar Archive Compression

# Понимание характера использования данных

## Температурная карта данных ('heat map')

0100110101001101	10010111101010101	10010111101010101
00110101111010101	0101010101010101	0101010101010101
0111010101010110	0111010101010110	0111010101010110
1001000101010101	1001000101010101	1001000101010101
1001011101010101	1101000101011101	1001011101010101
0101010101010101	0101110101011101	0101010101010101
0111110100011110	0111110100011110	0111110100011110
1001000101010101	1001000101010101	1001000101010101
1011011100010101	1011011100010101	1011011100010101
0101010101010101	0101010101010101	0101010101010101
0101110000011111	0101110000011111	1001000101010101
1001000101010101	1001000101010101	1001000101010101
1001000101010101	10010111101010101	0111110101011111
0101010101010101	1101010101010101	1101010101010101
1001000101010101	0111110101011111	0111110101011111
1001100101010101	1001100101010101	1001100101010101

# Понимание характера использования данных

Температурная карта данных ('heat map')



0100110101001101	1001011101010101	1001011101010101
0011010111010101	0101010101010101	0101010101010101
0111010101010110	0111010101010110	0111010101010110
1001000101010101	1001000101010101	1001000101010101
1001011101010101	1101000101011101	1001011101010101
0101010101010101	0101110101011101	0101010101010101
0111110100011110	0111110100011110	0111110100011110
1001000101010101	1001000101010101	1001000101010101
1011011100010101	1011011100010101	1011011100010101
0101010101010101	0101010101010101	0101010101010101
0101110000011111	0101110000011111	1001000101010101
1001000101010101	1001000101010101	1001000101010101
1001000101010101	1001011101010101	0111110101011111
0101010101010101	1101010101010101	1101010101010101
1001000101010101	0111110101011111	0111110101011111
1001100101010101	1001100101010101	1001100101010101

# Автоматическая оптимизация данных

Сжатие на основе характера использования данных

01110101010010

10000100010101

**“Горячие”  
данные**

10101010111010100110101  
11000010100010110111010  
10100101001001000010001  
01010110100101101001110  
00010100100101000010010  
00010001010101110011010  
10100101001001000010001

**3X**

Advanced Row Compression

**“Теплые”  
данные**

1010101011101010011010111000010100  
01011011101010100101001000100001000  
1010101101001011010011100001010010  
0101000010010000100010101011010010  
1000010100100101001010111000010  
1110010100100101001010110111011010

**10X**

Columnar Query Compression

**Архивные  
данные**

10101010111010100110101110000101000101011  
101010100101001001000010001010101101001011  
010011100001010010010100001001000010001010  
101010101110101001101011100001010001011011  
101010101110101001101011100001011101011001

**15X**

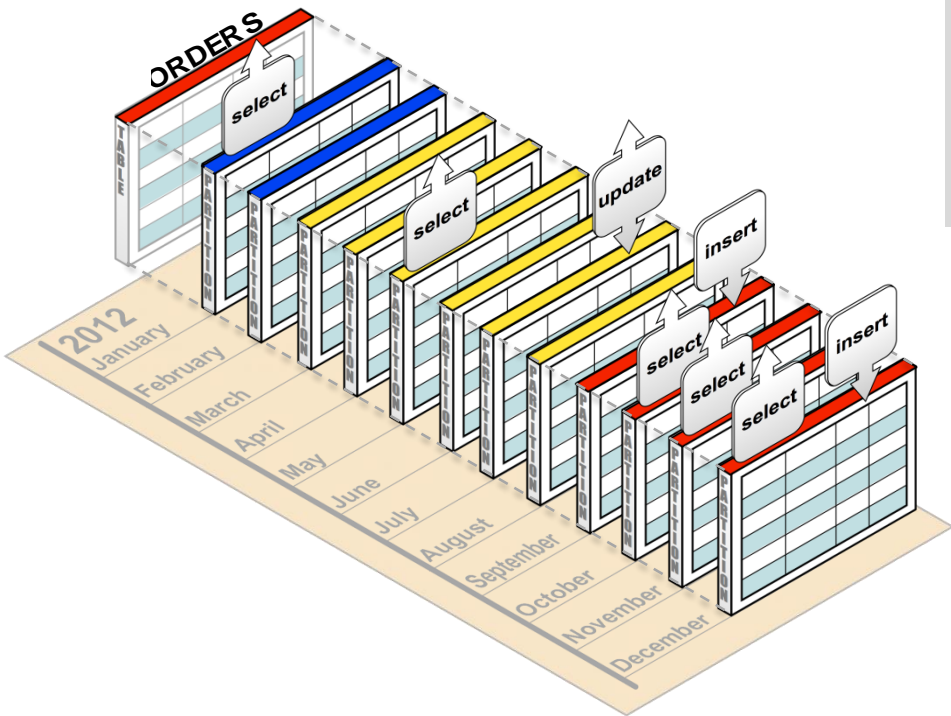
Columnar Archive Compression

ORACLE



# Сжатие на уровне строк или секций таблицы

В зависимости от статистики использования – в “фоне”



```
SQL> ALTER TABLE EMPLOYEE ILM  
ADD POLICY
```

```
ROW STORE COMPRESS ADVANCED ROW  
AFTER 1 DAY OF NO MODIFICATION;
```

```
SQL> ALTER TABLE ORDERS ILM ADD  
POLICY
```

```
COMPRESS ADVANCED SEGMENT  
ON ORDERS_CLOSED ;
```

# Oracle Database In-Memory Option



ORACLE

# Противоречие которое длится десятилетия

## Строчный формат и поколоночный

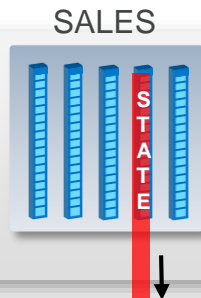
Row



- OLTP-операции работают быстрее со строчным форматом

- Быстрая обработка нескольких строк, много колонок

Column



- Аналитика работает быстрее с колоночным форматом

- Отчет о сумме продаж по штату
- Быстрая обработка нескольких колонок, много строк

Приходится выбирать один формат и идти на компромиссы

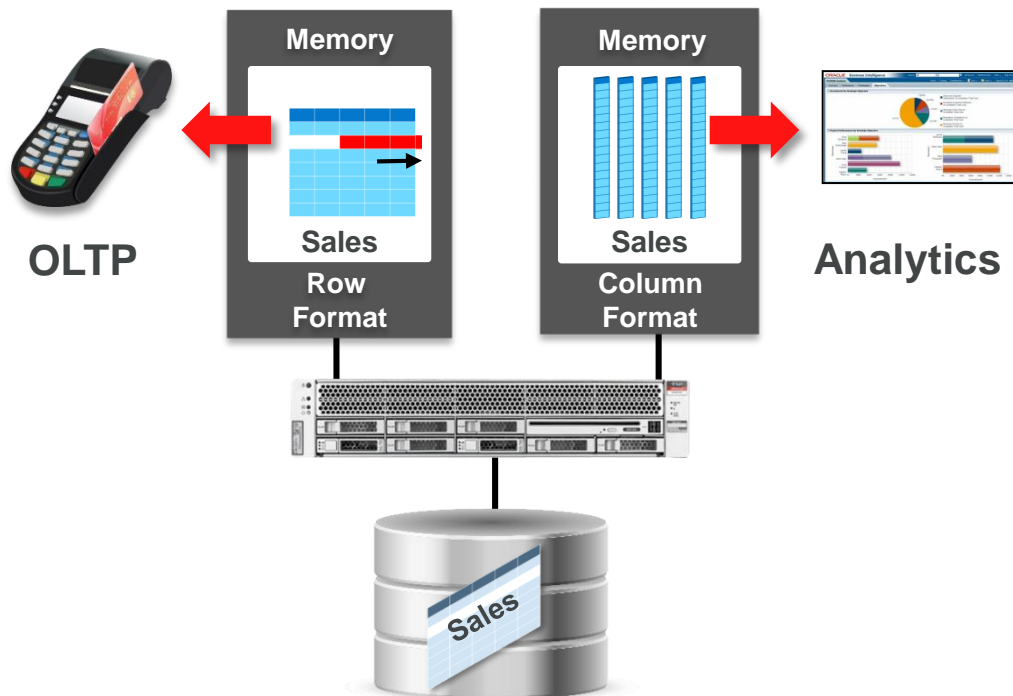
# Oracle Database In-Memory Option

- **Мощная технология обработки данных в памяти**
  - Эффективно интегрирована в СУБД Oracle Database
- **Обеспечивает увеличение производительности**
  - Аналитических и Ad-Нос отчетов на “живых” данных
  - Для OLTP-приложений и хранилищ данных
- **Готовое решение в “коробке”: прозрачно для приложений!**

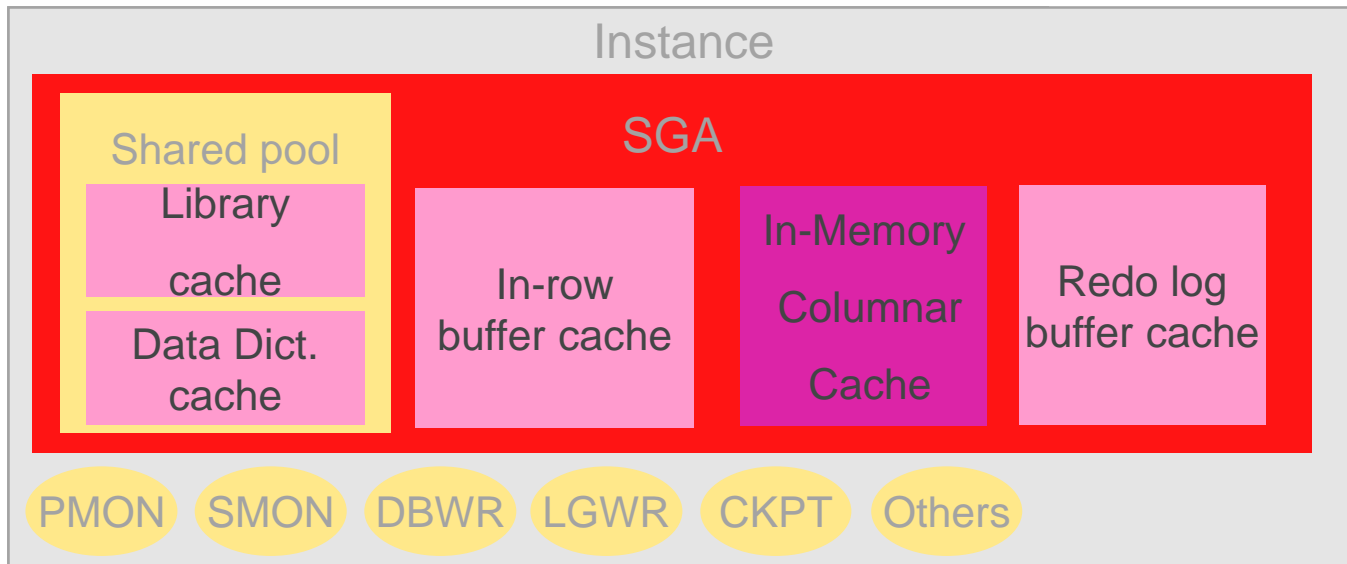


# In-Memory Option: Оба формата в памяти СУБД

- И строчный и поколоночный формат хранения в памяти для одних и тех же данных/таблиц
- Данные одновременно активны и транзакционно согласованы
- В 100 раз быстрее аналитика & отчетность: поколоночный формат
- В 2 раза быстрее OLTP: строчный формат



# Отдельный кэш в SGA

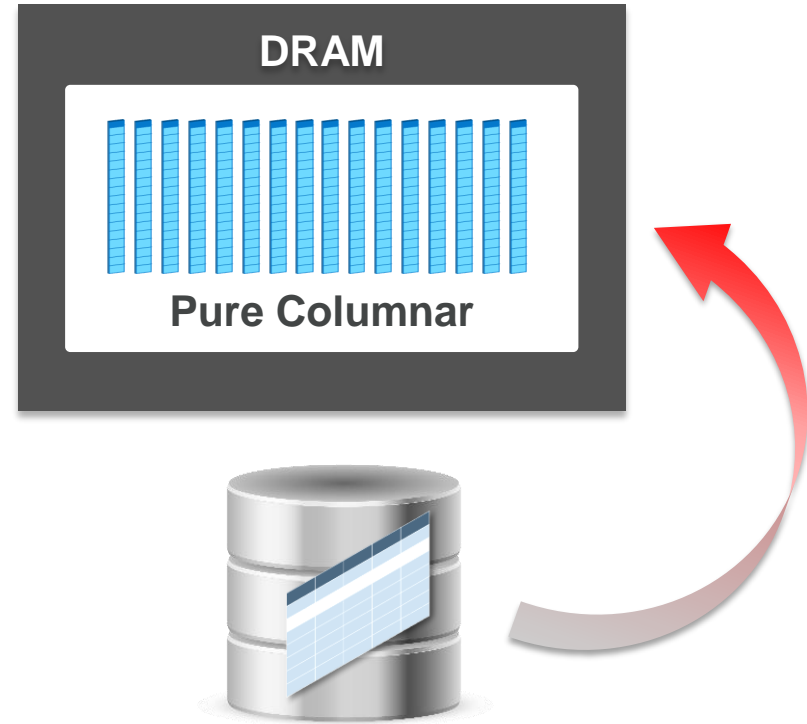


- Динамический параметр INMEMORY\_SIZE

```
SQL> ALTER SYSTEM SET inmemory_size=100G SCOPE=BOTH;
```

# Oracle In-Memory Columnar Cache

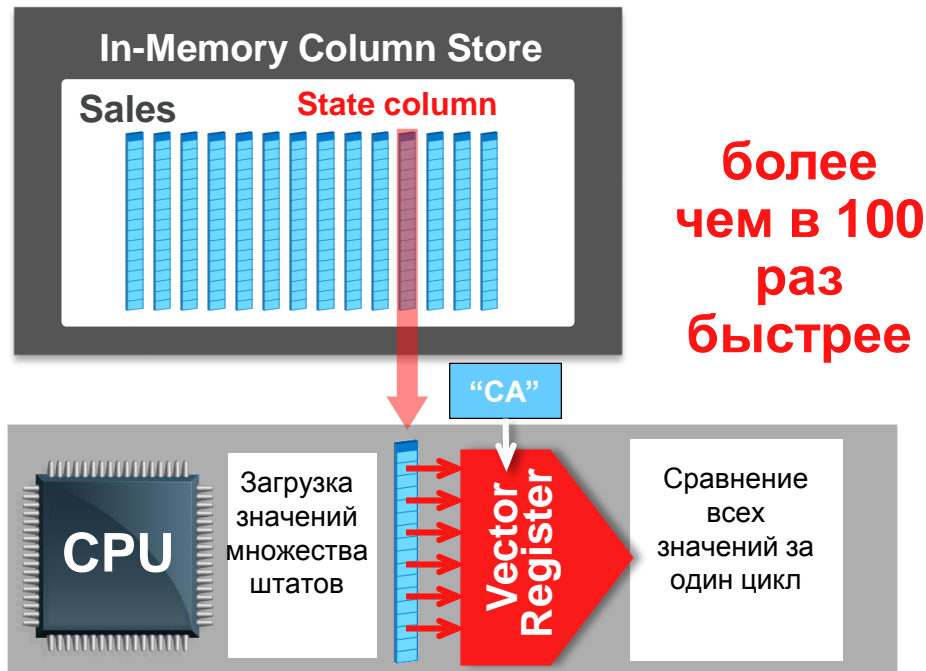
- Данные в представлении по колонкам без генерации redo-информации
- Минимальные затраты на изменения – даже для OLTP-транзакций
- Данные загружаются в кэш при первом обращении



# Сканирование миллиарда строк в секунду на процессорном ядре

**Пример:** Найти все продажи в штате CA

- Каждое процессорное ядро сканирует одну колонку
- При сканировании используются быстрые векторные SIMD-инструкции
- **Миллиарды строк в секунду** сканируются одним ядром





# Изменение плана запроса

SQL-оптимизатор перестраивает план запроса

```
SQL> SELECT count(*) FROM cities;
```

Execution Plan

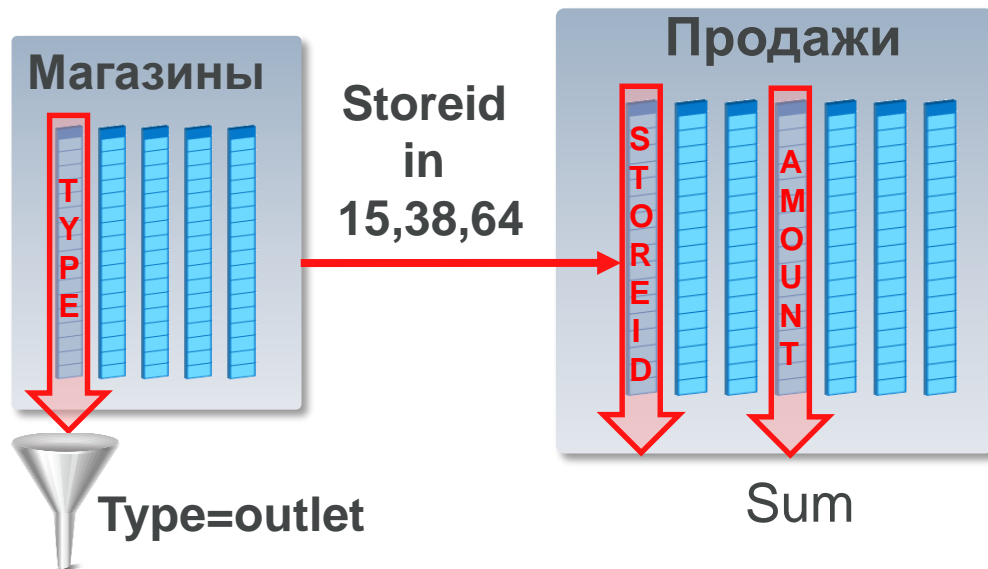
Plan hash value: 2756775702

Id	Operation	Name	Rows	Cost (%CPU)
0	SELECT STATEMENT		1	0 (0)
1	SORT AGGREGATE		1	
2	TABLE ACCESS INMEMORY FULL	CITIES	1	

# Сканирование и объединение данных из нескольких таблиц

- Конвертирует join в быстрые сканы колонок
- Joins выполняются в **10 раз** быстрее

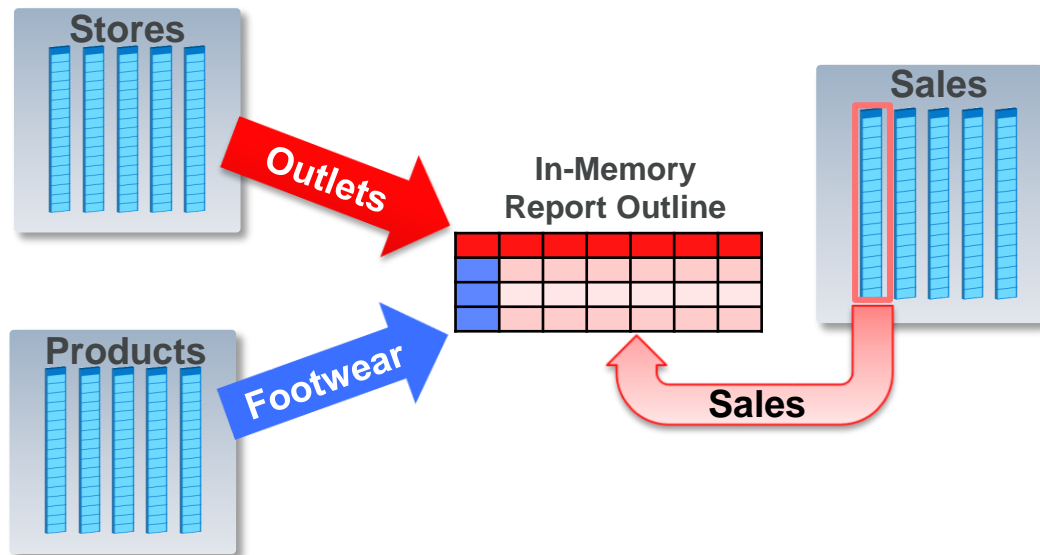
**Пример:** Найти все продажи в outlet-магазинах



# Мгновенная генерация отчетов: доли секунд

**Пример:** Показать тренды продаж обуви в outlet магазинах

- Динамически создает в памяти объект-отчет
- Объект заполняется во время сканирования таблицы продаж
- Отчеты строятся в **20 раз** быстрее без заранее созданных кубов



# Сжатие столбцов в columnar-кэше

## Методы сжатия

Метод сжатия	Синтаксис	Описание
<ul style="list-style-type: none"><li>BASIC</li></ul>	MEMCOMPRESS BASIC	<ul style="list-style-type: none"><li>Данные не сжимаются</li></ul>
<ul style="list-style-type: none"><li>THROUGHPUT (default)</li></ul>	MEMCOMPRESS FOR QUERY	<ul style="list-style-type: none"><li>Минимальная степень сжатия. Высокая производительность для запросов. Метод по умолчанию.</li></ul>
<ul style="list-style-type: none"><li>BALANCED</li></ul>	MEMCOMPRESS FOR CAPACITY LOW	<ul style="list-style-type: none"><li>Средняя степень сжатия. Самая высокая производительность для отчетов.</li></ul>
<ul style="list-style-type: none"><li>SPACE</li></ul>	MEMCOMPRESS FOR CAPACITY HIGH	<ul style="list-style-type: none"><li>Высокая степень сжатия. Хорошая производительность для отчетов.</li></ul>

# Управление приоритетом загрузки в кэш

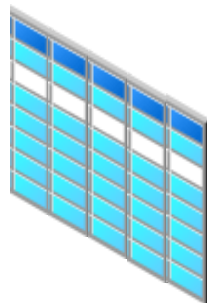
Приоритет	Синтаксис	Описание
<ul style="list-style-type: none"><li>NONE (default)</li></ul>	PRIORITY NONE	<ul style="list-style-type: none"><li>Приоритет загрузки определяется СУБД. Загрузка может быть задержана, если память используется для другого более приоритетной задачи. Запрос переключается на обычный (in-row) кэш.</li></ul>
<ul style="list-style-type: none"><li>LOW</li></ul>	PRIORITY LOW	<ul style="list-style-type: none"><li>Низкий приоритет</li></ul>
<ul style="list-style-type: none"><li>MEDIUM</li></ul>	PRIORITY MEDIUM	<ul style="list-style-type: none"><li>Средний приоритет</li></ul>
<ul style="list-style-type: none"><li>HIGH</li></ul>	PRIORITY HIGH	<ul style="list-style-type: none"><li>Высокий приоритет</li></ul>
<ul style="list-style-type: none"><li>CRITICAL</li></ul>	PRIORITY CRITICAL	<ul style="list-style-type: none"><li>Высший приоритет</li></ul>

# OLTP работает медленно из-за аналитических индексов

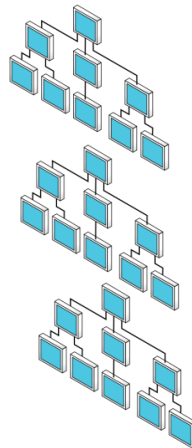
- Большинство индексов в OLTP (например, в ERP) базах строится только для аналитических запросов
- Индексы хорошо подходят для предсказуемых запросов (и в памяти, и на диске)
- Вставка одной строки в таблицу приводит к обновлению 10-20 аналитических индексов:

**Медленно!**

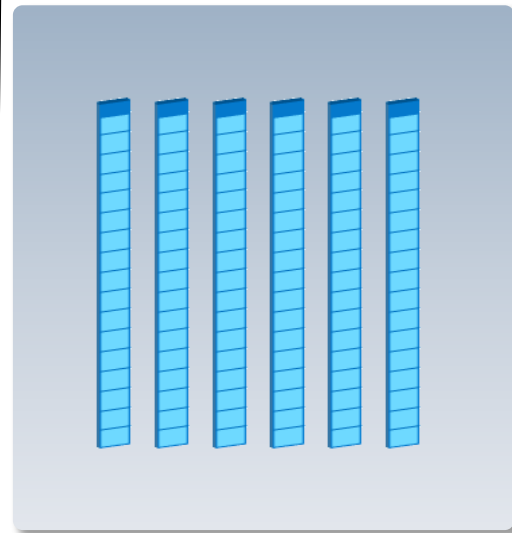
Таблица



1 - 3  
OLTP  
индекса

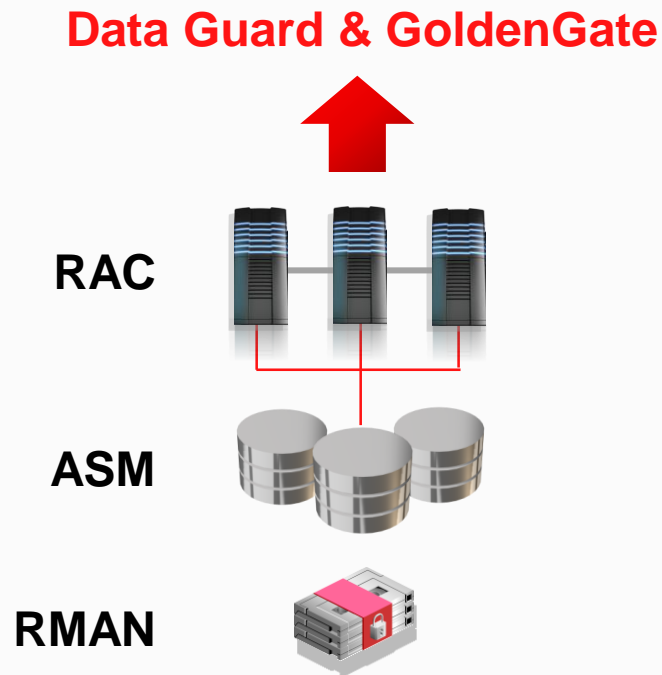


**Поколоночное  
хранение в памяти**



# Oracle In-Memory – высокая доступность

- Представление по столбцам в памяти не влияет на формат данных на диске: datafiles, logging, backup, recovery, и т.д.
- Все технологии, в том числе ASM, RAC, DG, GG работают прозрачно для In-Memory Option
- Защита от любых видов ошибок
  - На уровне железа
  - Логические ошибки приложения (Flashback)

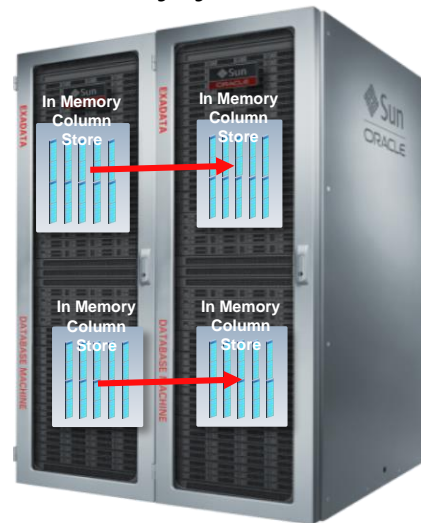


# Oracle In-Memory в RAC

- Управление распределением объектов в кэше между узлами RAC:
  - AUTO DISTRIBUTE – синхронизацией кэша управляет СУБД (поведение по умолчанию)
  - DUPLICATE – кэши принудительно синхронизируются между узлами RAC
  - Задается на уровне всей таблицы

```
SQL> ALTER TABLE cites INMEMORY  
      DUPLICATE;
```

```
Table altered.
```



ORACLE



# Полный синтаксис кэширования таблицы

Мощный и гибкий синтаксис

```
SQL> ALTER TABLE cities
INMEMORY
  PRIORITY CRITICAL
  DUPLICATE
  INMEMORY MEMCOMPRESS FOR CAPACITY HIGH (Country_Id, Time_Zone)
  INMEMORY MEMCOMPRESS BASIC (Id, Name, Name_Eng)
  NO INMEMORY (Created, Modified, State);
```

Table altered.

# Storage Index в In-Memory Columnar Store

- Хранит минимальное и максимальное значение столбца в каждом экстенсте памяти кэша
- Прозрачно исключает ненужные сканирования столбцов, например:  
*WHERE prod\_id > 14  
AND prod\_id < 29*

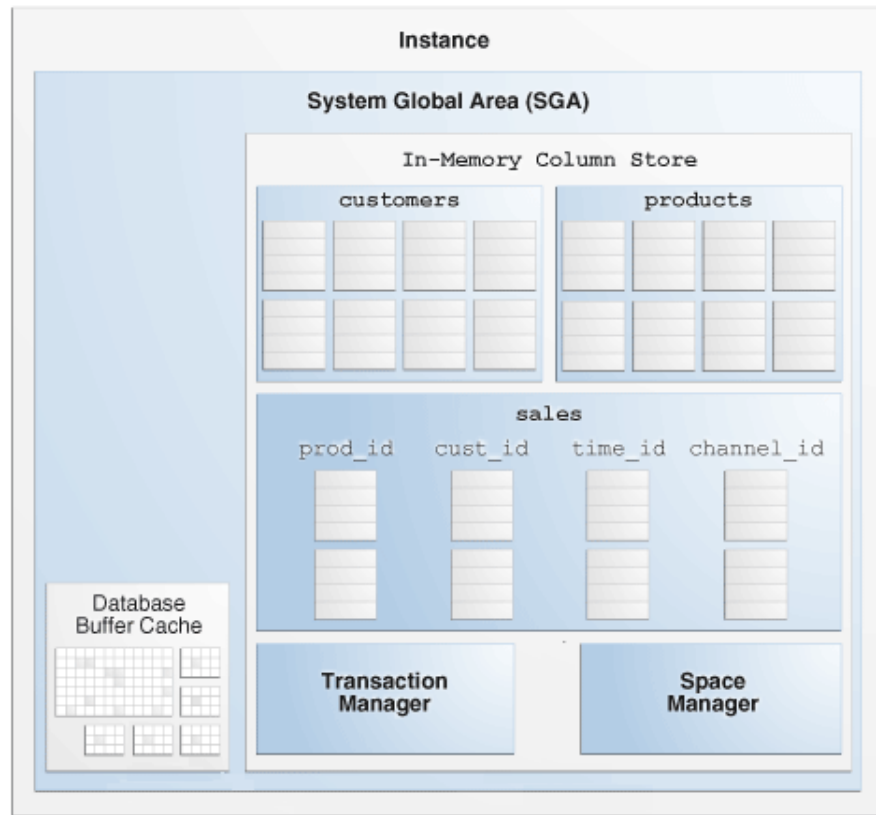
<del>1</del>	15
<del>2</del>	16
<del>3</del>	17
<del>4</del>	18
<del>5</del>	19
<del>6</del>	20
<del>7</del>	21
<del>8</del>	22
<del>9</del>	23
<del>10</del>	24
<del>11</del>	25
<del>12</del>	26
<del>13</del>	27
<del>14</del>	28

## Storage Index

```
ext 1:  min 1, max 7
ext 2:  min 8, max 14
ext 3:  min 15, max 21
ext 4:  min 22, max 28
```

# In-Memory Option - архитектура

- Space Manager
  - Управляет памятью в In-Memory Column Store (create, extend, drop)
  - Загружает данные в кэш
- Transaction Manager
  - Обеспечивает согласованность данных в In-Memory Column Store с буферным кэшем
  - Обеспечивает версиюность



# Oracle In-Memory Options не требует изменения приложений

Полная функциональность	- Нет ограничений на SQL
Простота реализации	- Не нужна миграция данных
Полная совместимость	- Не надо изменять приложения
Полностью Multitenant	- Oracle In-Memory готова для cloud

**ORACLE®**  
E-BUSINESS SUITE

**ORACLE®**  
FUSION APPLICATIONS

**ORACLE®**  
JD EDWARDS

**ORACLE®**  
PEOPLESOFT

**ORACLE®**  
SIEBEL

Приложения получают все преимущества In-Memory опции без изменения кода приложения

ORACLE®

# Oracle Database In-Memory и TimesTen

## TimesTen In-Memory Database

- Встраиваемая БД для приложений
- Экстремальная быстрая производительность для OLTP-нагрузки (время ответа - микросекунды)
- Адаптивный кэш для инженерной системы Oracle Exalytics



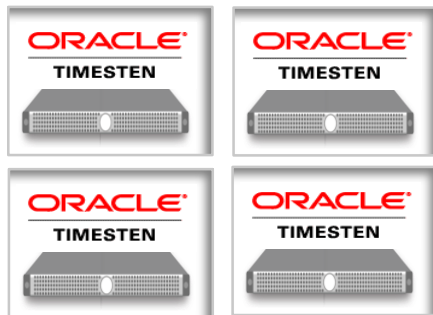
## Oracle Database In-Memory Option

- Масштабируемая обработка данных в памяти для любых видов нагрузки, в том числе и смешанной (OLTP и DWH)

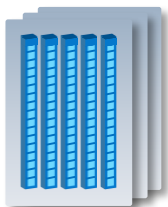


# TimesTen In-Memory Database и Columnar

## TimesTen Grid



## Scale-Out In-Memory DB

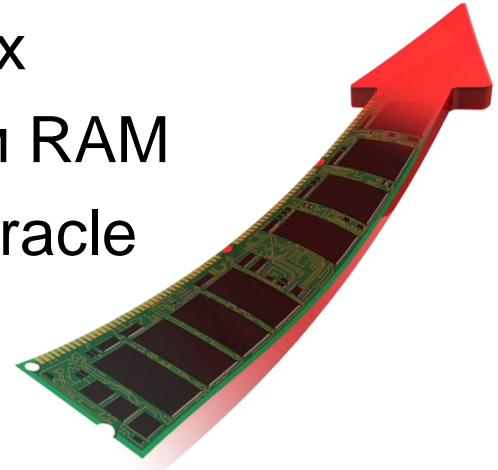


## Columnar

- TimesTen также используется и для аналитических задач
- TimesTen имеет технологию обработки по столбцам для ускорения аналитики
  - Ядро системы обработки и хранения данных по столбцам общее для ТТ и Oracle Database
  - Дополнительно ТТ имеет встроенный аналитический “движок”

# Заключение: Oracle In-Memory Option

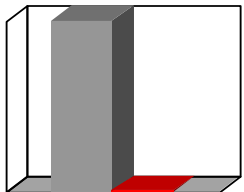
- Экстремальная производительность для OLTP и хранилищ данных на актуальных данных
- Прозрачное масштабирование по CPU и RAM
- Прозрачно для всех технологий СУБД Oracle



**Все преимущества обработки в памяти  
без изменения приложений**

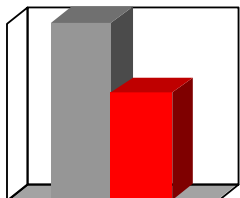
# Увеличение производительности

## Database Creation



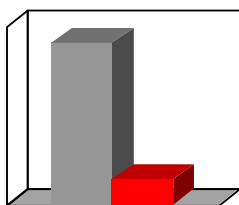
Up to 1000x faster

## Database Upgrade



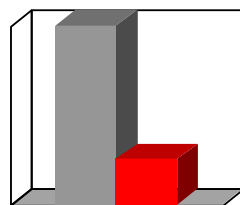
Up to 40% faster

## Adaptive Query Optimization



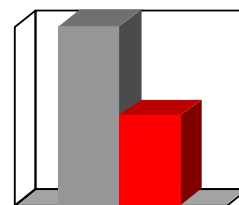
Up to 6x faster

## Parallel Execution



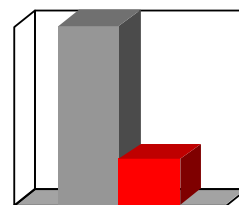
Up to 4x faster

## Large Parallel Sorts



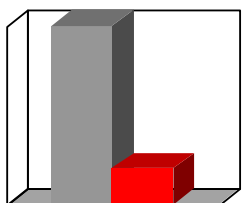
Up to 2x faster

## Parallel Union All



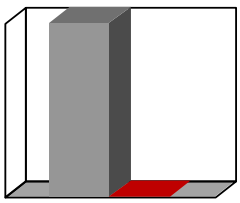
Up to 4x faster

## Materialized View Refresh



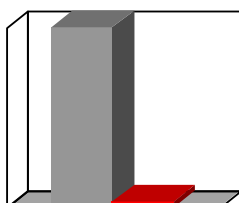
Up to 5x faster

## Asynchronous Global Indexes



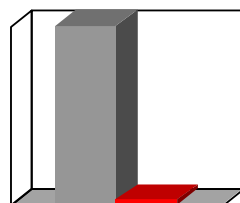
Instant partition maintenance

## String operations in PL/SQL



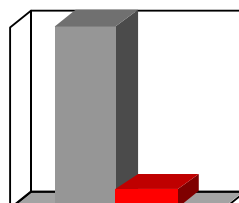
Up to 30x faster

## Spatial Queries



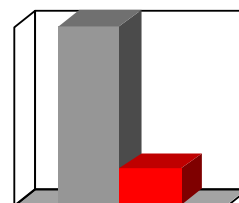
Up to 40x faster

## Network Encryption



Up to 10x faster

## Auditing



Up to 5x faster

ORACLE



# Включена в облако

**ORACLE<sup>®</sup>**  

---

**DATABASE** **12<sup>c</sup>**

Application Development

Big Data

Consolidation

Data Optimization

Data Warehousing

High Availability

In-Memory

Performance & Scalability

Security & Compliance

**ORACLE<sup>®</sup>**

**Hardware and Software**

**ORACLE®**

**Engineered to Work Together**

ORACLE®