

# **Администрирование баз данных и приложений**

**Введение в СУБД ORACLE 12с**

**Лекция 1**

# До того, как начать

---

- ▶ 34 часа лекций, 51 час лабораторных работ
- ▶ 5 семестр ПОИТ, 6 семестр ИСИТ - курсовой проект, построение всей необходимой инфраструктуры базы данных, самостоятельное изучение и исследование технологий СУБД
- ▶ Версия СУБД: Oracle 12c
- ▶ Лабораторные работы в 309-1, виртуальные серверы (VMware ESX) Oracle 12c, на компьютерах установлены клиентские компоненты СУБД и Oracle IDE
- ▶ Все продукты Oracle доступны для скачивания на сайте [www.oracle.com](http://www.oracle.com)
- ▶ Задания на лабораторные работы и презентации лекций diskstation/Преподаватели/Блинова
- ▶ Рефераты и доклады



# Администрирование баз данных

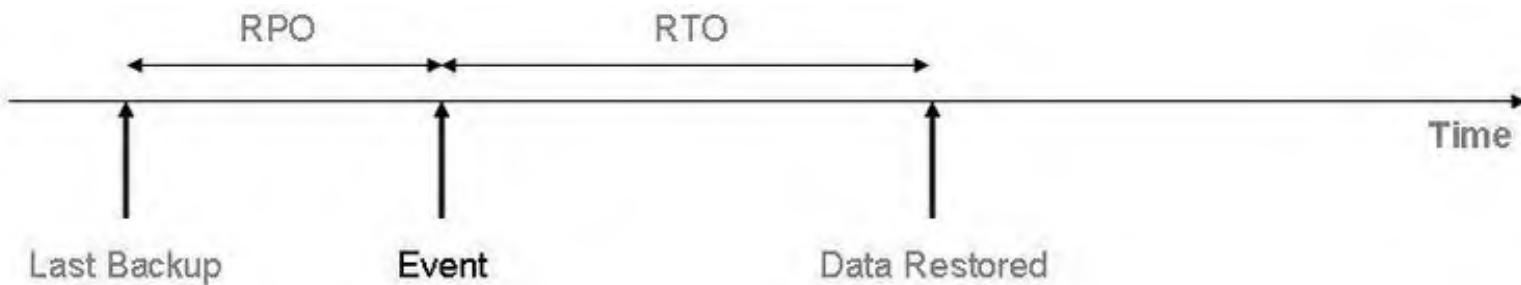
---

- ▶ Проектирование базы данных
- ▶ Восстановление работоспособности
- ▶ Обеспечение целостности
- ▶ Обеспечение безопасности в базе данных
- ▶ Оптимизация и поддержка производительности
- ▶ Мониторинг системы
- ▶ Обеспечение перехода на новую версию



# Понятия RTO и RPO

- ▶ RTO - Время восстановления - recovery time objective - допустимое время простоя сервиса в случае сбоя
- ▶ RPO - Точка возврата - recovery point objective - допустимый объем возможных потерь данных в случае сбоя



# Oracle Corporation

---

- ▶ Создана в 1977г.
- ▶ имеет подразделения в 145 странах
- ▶ вторая после Microsoft по прибыльности компания – поставщик программного обеспечения
- ▶ финансовые показатели в 2019 году: оборот \$39,51 млрд, чистая прибыль \$14,8 млрд
- ▶ количество сотрудников в 2019 году – 136 000
- ▶ 30% глобального рынка программного обеспечения
- ▶ основные продукты Oracle RDBMS, TimesTen, Oracle Enterprise Linux (на основе Red Hat), JDeveloper, WebLogic, Apex
- ▶ поглощение Sun Microsystems: MySQL, Java



# СУБД ORACLE - объектно-реляционная система управления базами данных

---

- ▶ 1977: первая версия
- ▶ 2005: Oracle 10g Release 2
- ▶ 2009: Oracle 11g Release 2 поддержка на платформах Windows (с Windows Server 2008), Linux, Solaris,
- ▶ 2013: Oracle 12c
- ▶ 2018: Oracle 18c
  - ▶ поддержка JSON
  - ▶ поддержка длинных наименований для объектов БД
  - ▶ PL/SQL-функции в выражении WITH
  - ▶ PTF (Polymorphic Table Functions)
  - ▶ Private Temporary Tables
- ▶ 2019: Oracle 19c
  - ▶ автоматическая индексация
  - ▶ гибридные секционированные таблицы
  - ▶ карантин запросов

# Основные редакции СУБД ORACLE 12с

---

- ▶ Express Edition: бесплатная редакция
- ▶ Standard Edition One: начальная редакция для дома и малого бизнеса
- ▶ Standard Edition: для малого и среднего бизнеса
- ▶ Enterprise Edition: решения промышленного уровня



# Лицензирование

---

- ▶ Named User Plus – по пользователям, можно конкретно подсчитать точное количество пользователей системы
- ▶ Processor – по процессорной мощности серверов на неограниченное количество пользователей



# Лицензирование

## Oracle Database 12c Enterprise Edition. Лицензия

**Внимание!** Продукт без техподдержки не поставляется.

1	<a href="#">Лицензия на именованного пользователя</a> <a href="#">лицензия</a>	<input type="button" value="25"/> <input type="button" value="▲"/> <input type="button" value="▼"/>	1 843 575 руб. 73 743 руб. — цена за 1 лицензию	<a href="#">В корзину</a>
2	<a href="#">Процессорная лицензия</a> <a href="#">лицензия</a>	<input type="button" value="1"/> <input type="button" value="▲"/> <input type="button" value="▼"/>	3 687 127 руб.	<a href="#">В корзину</a>

\* Поставка ключа в электронном виде на e-mail, указанный при оформлении заказа. Срок доставки: от 3 раб.дн.

## Oracle Database 12c Enterprise Edition. Обновление и техподдержка лицензии на 1 год

Продукт приобретается одновременно с лицензией Oracle.

1	<a href="#">Лицензия на именованного пользователя</a> <a href="#">лицензия</a>	<input type="button" value="25"/> <input type="button" value="▲"/> <input type="button" value="▼"/>	486 700 руб. 19 468 руб. — цена за 1 лицензию	<a href="#">В корзину</a>
2	<a href="#">Процессорная лицензия</a> <a href="#">лицензия</a>	<input type="button" value="1"/> <input type="button" value="▲"/> <input type="button" value="▼"/>	973 402 руб.	<a href="#">В корзину</a>



# Дистрибутивы СУБД ORACLE

<http://www.oracle.com>

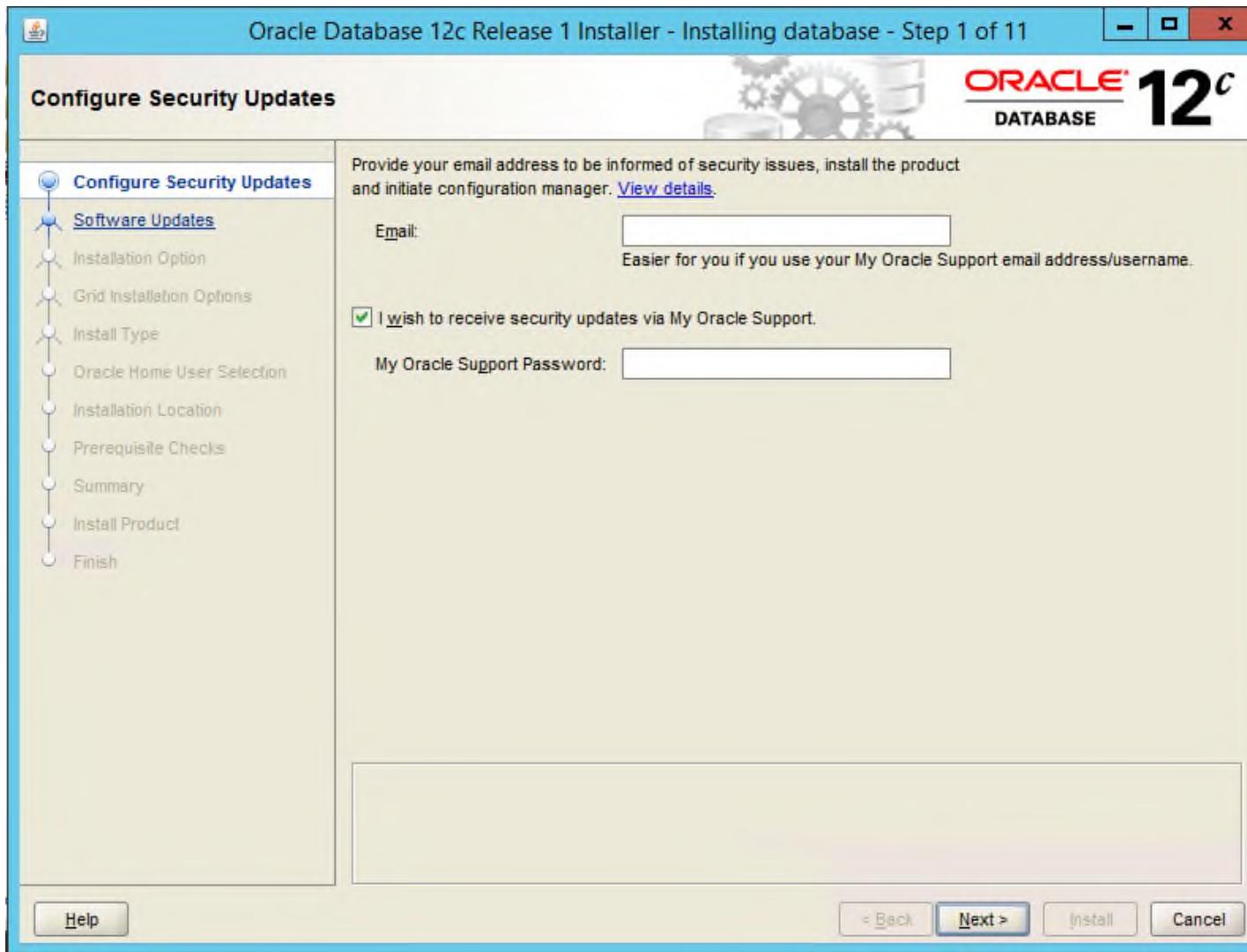
The screenshot shows the Oracle website's navigation bar at the top, featuring links for Sign In/Register, Help, Country, Communities, I am a..., I want to..., a search bar, and menu items for Products, Solutions, Downloads, Store, Support, Training, and Partners.

The main content area displays the "Oracle Database Software Downloads" section. It includes a note about accepting the OTN License Agreement and two radio buttons for "Accept License Agreement" and "Decline License Agreement".

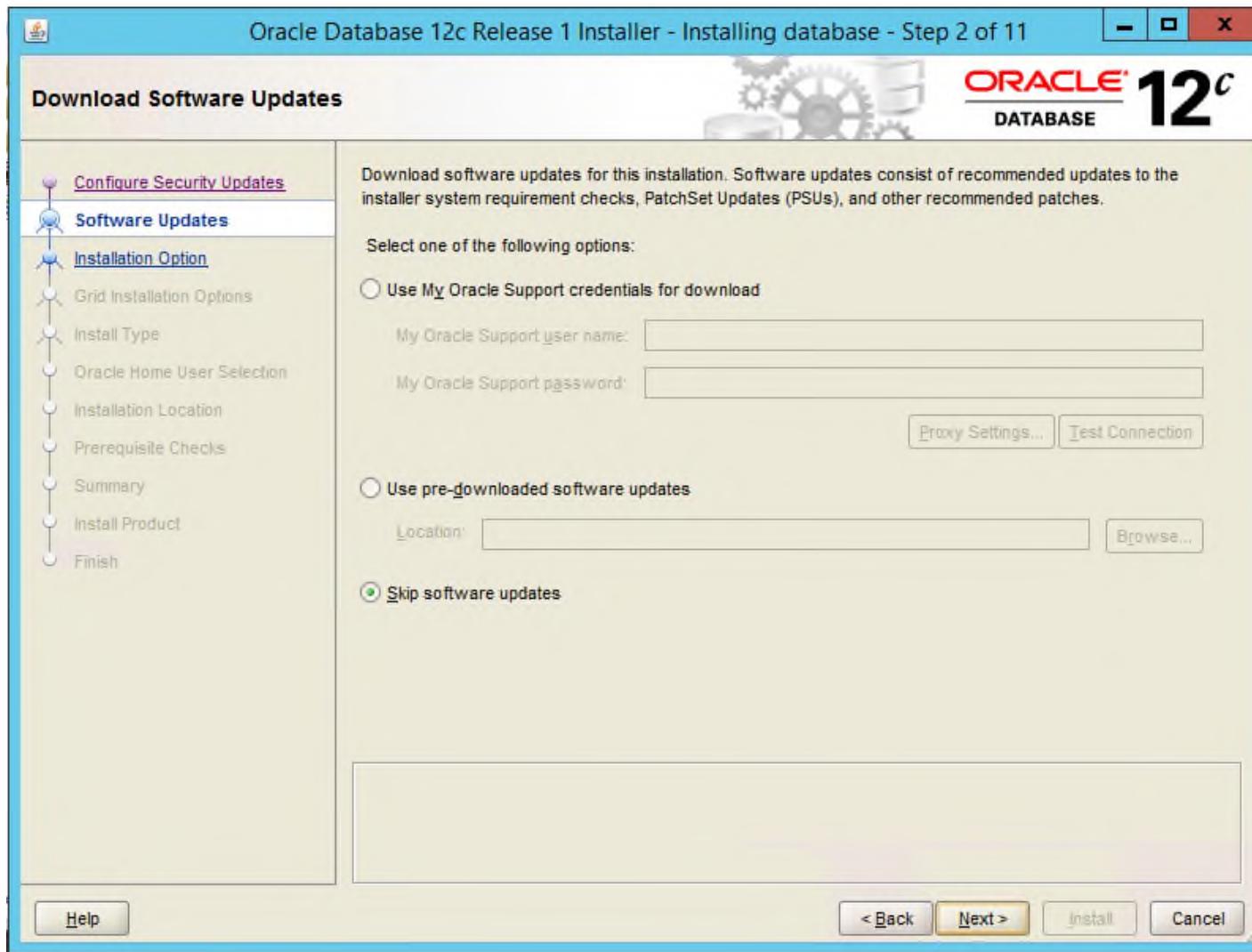
The "Oracle Database 12c Release 1" section lists download links for various platforms:

Platform	File 1	File 2	Size	Action
Microsoft Windows x64 (64-bit)	File 1	File 2	(2.5 GB)	See All
Linux x86-64	File 1	File 2	(2.5 GB)	See All
Oracle Solaris (SPARC systems, 64-bit)	File 1	File 2	(2.6 GB)	See All
Oracle Solaris (x86 systems, 64-bit)	File 1	File 2	(2.3 GB)	See All
HP-UX Itanium	File 1	File 2	(3.1 GB)	See All
AIX (PPC64)	File 1	File 2	(2.7 GB)	See All
zLinux64	File 1	File 2	(2.3 GB)	See All

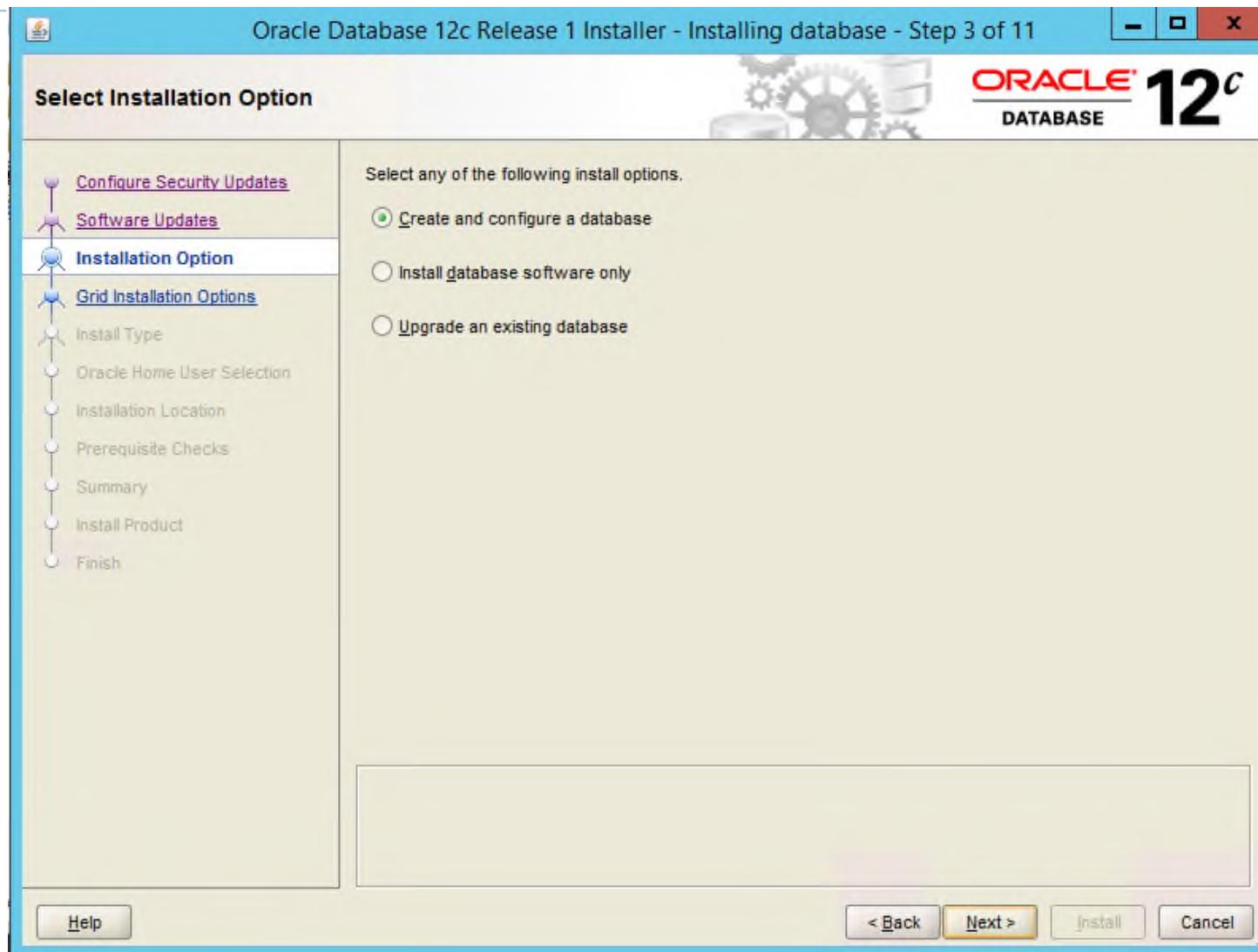
# Инсталляция ORACLE 12c EE



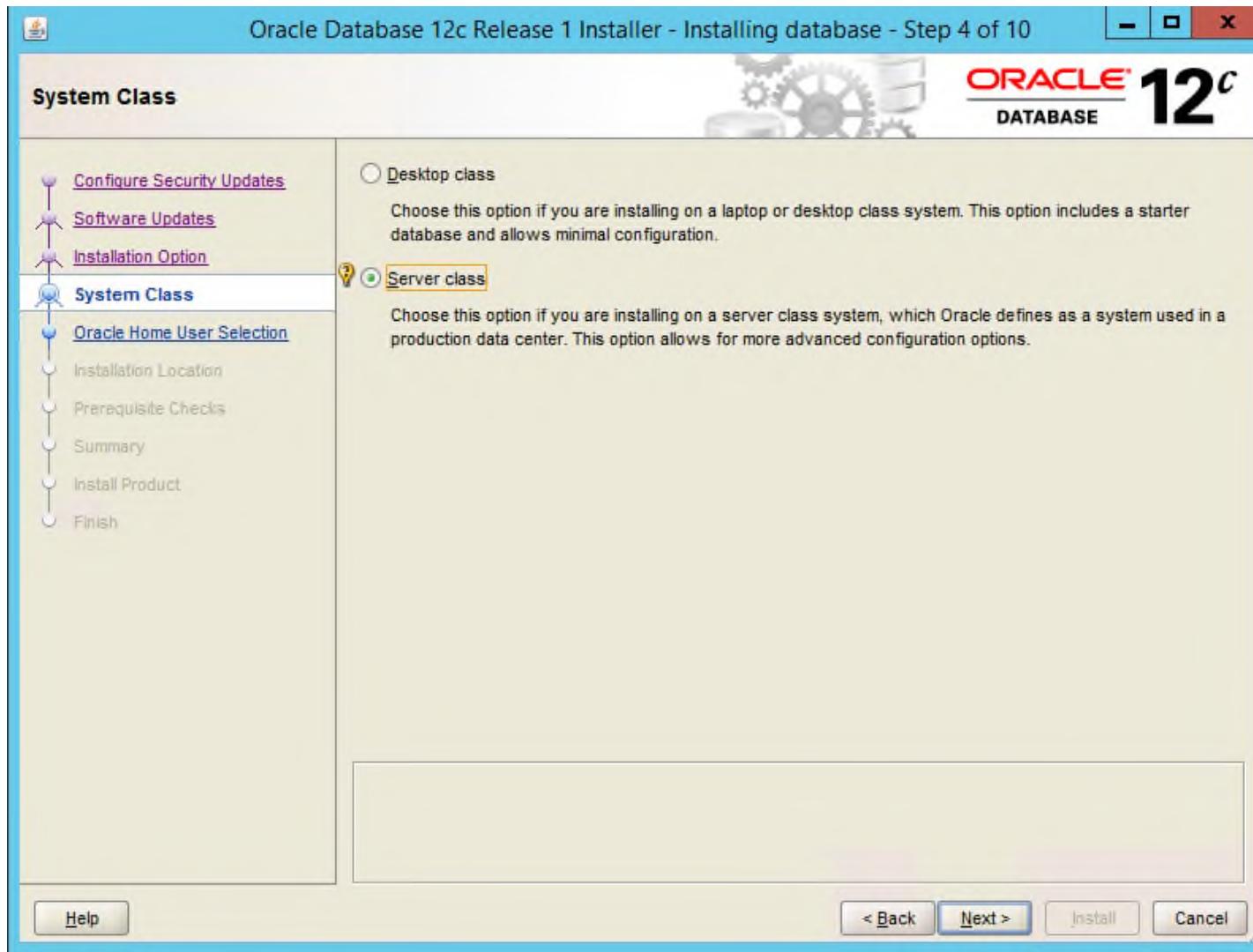
# Инсталляция ORACLE 12c EE



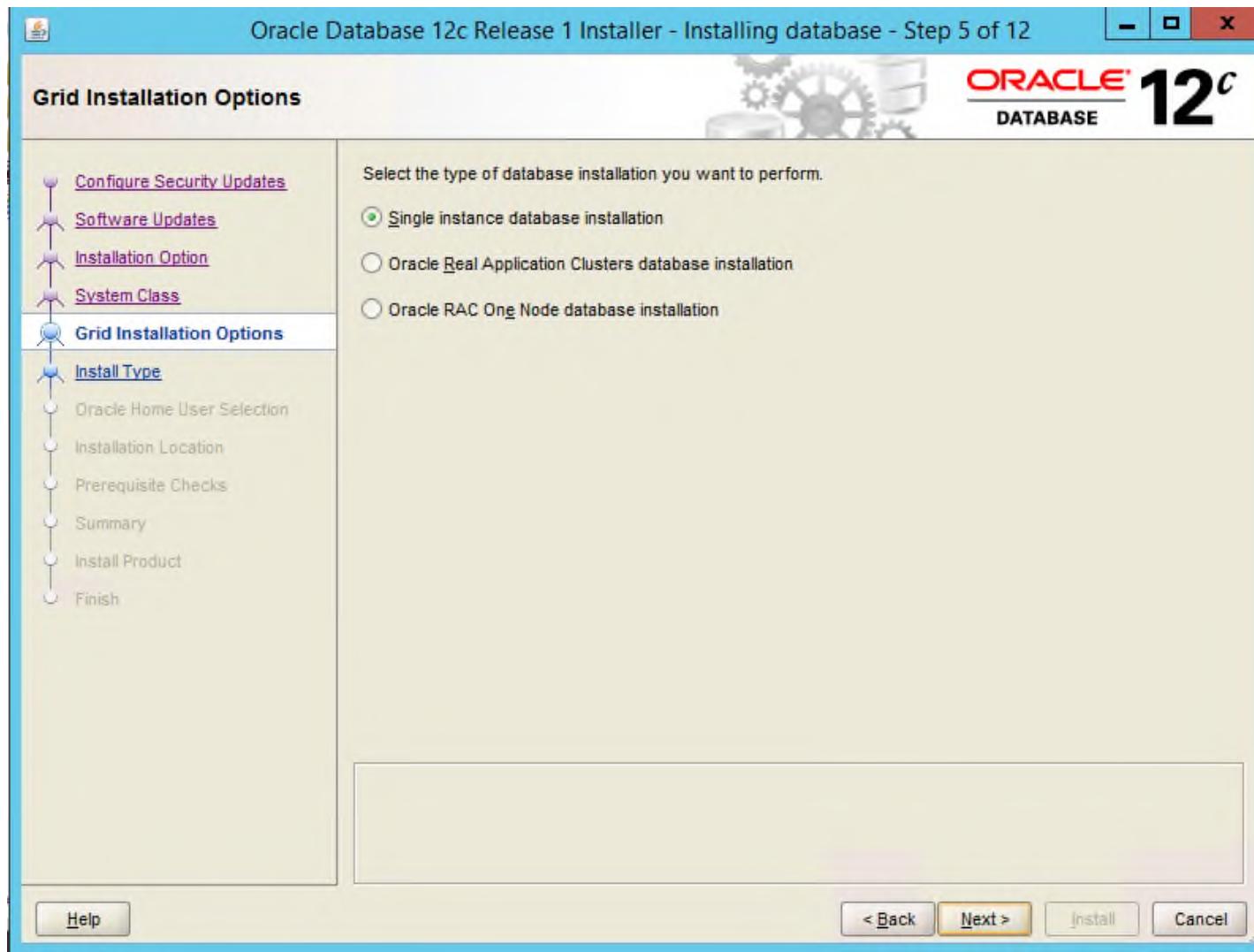
# Инсталляция ORACLE 12c EE



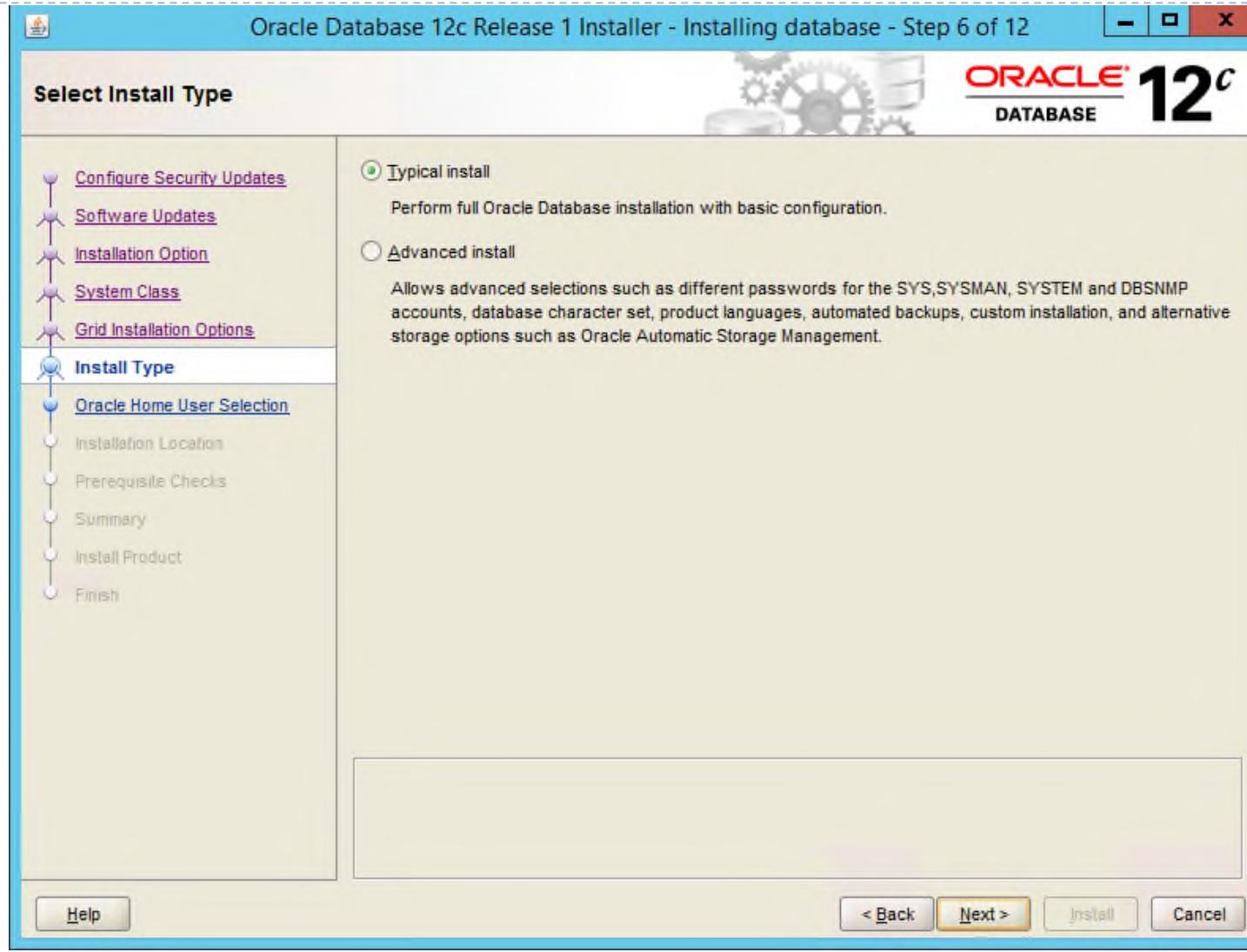
# Инсталляция ORACLE 12c EE



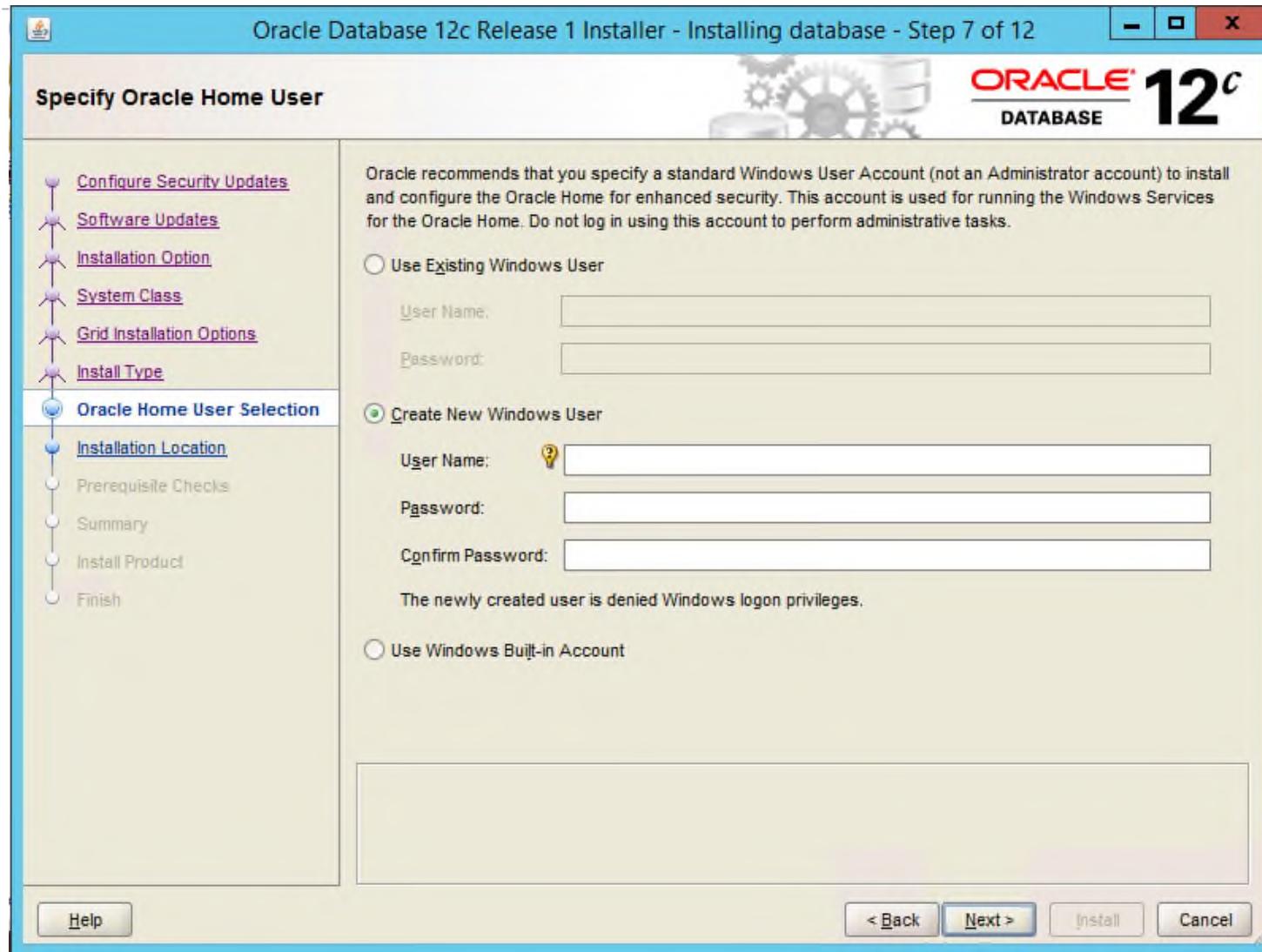
# Инсталляция ORACLE 12c EE



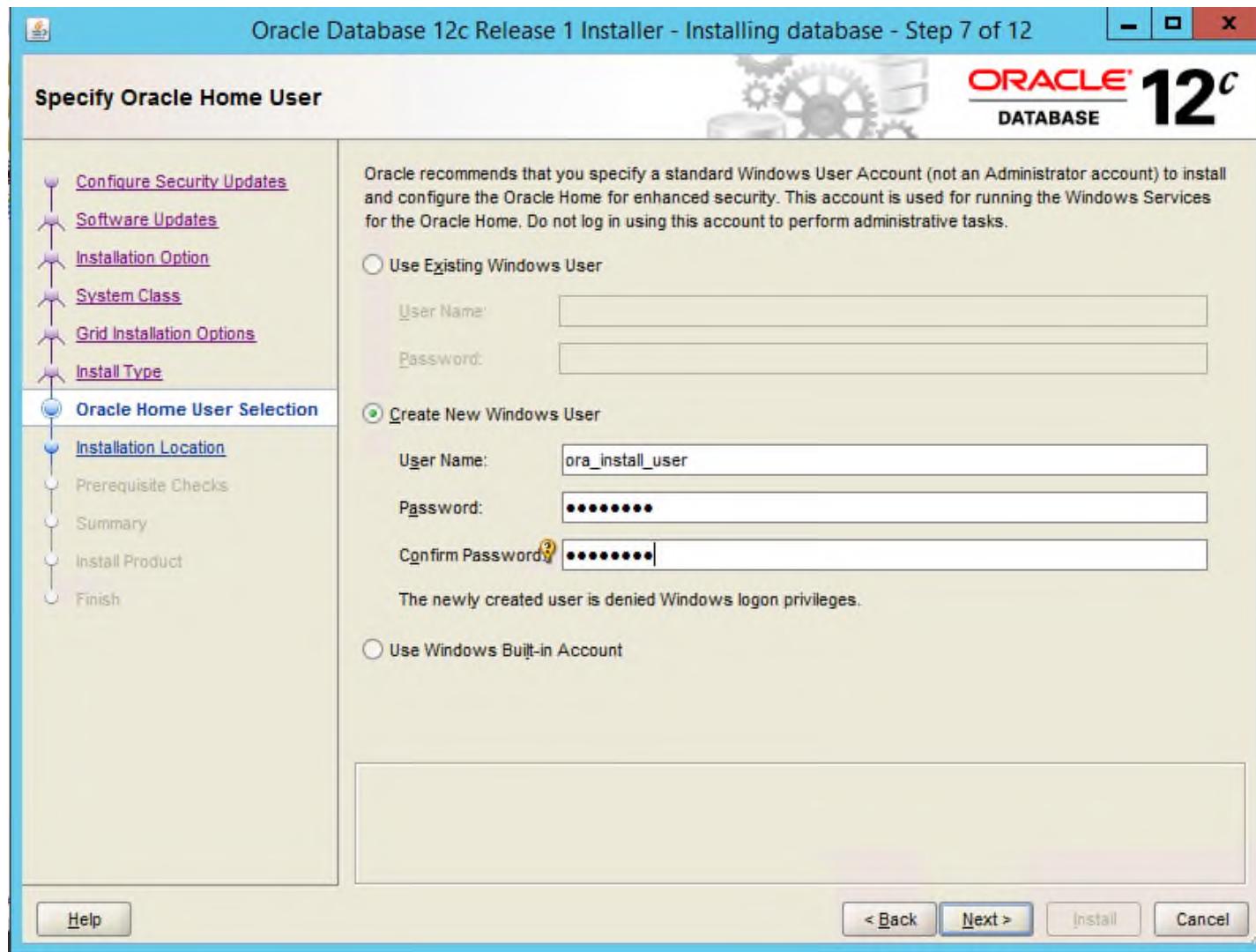
# Инсталляция ORACLE 12c EE



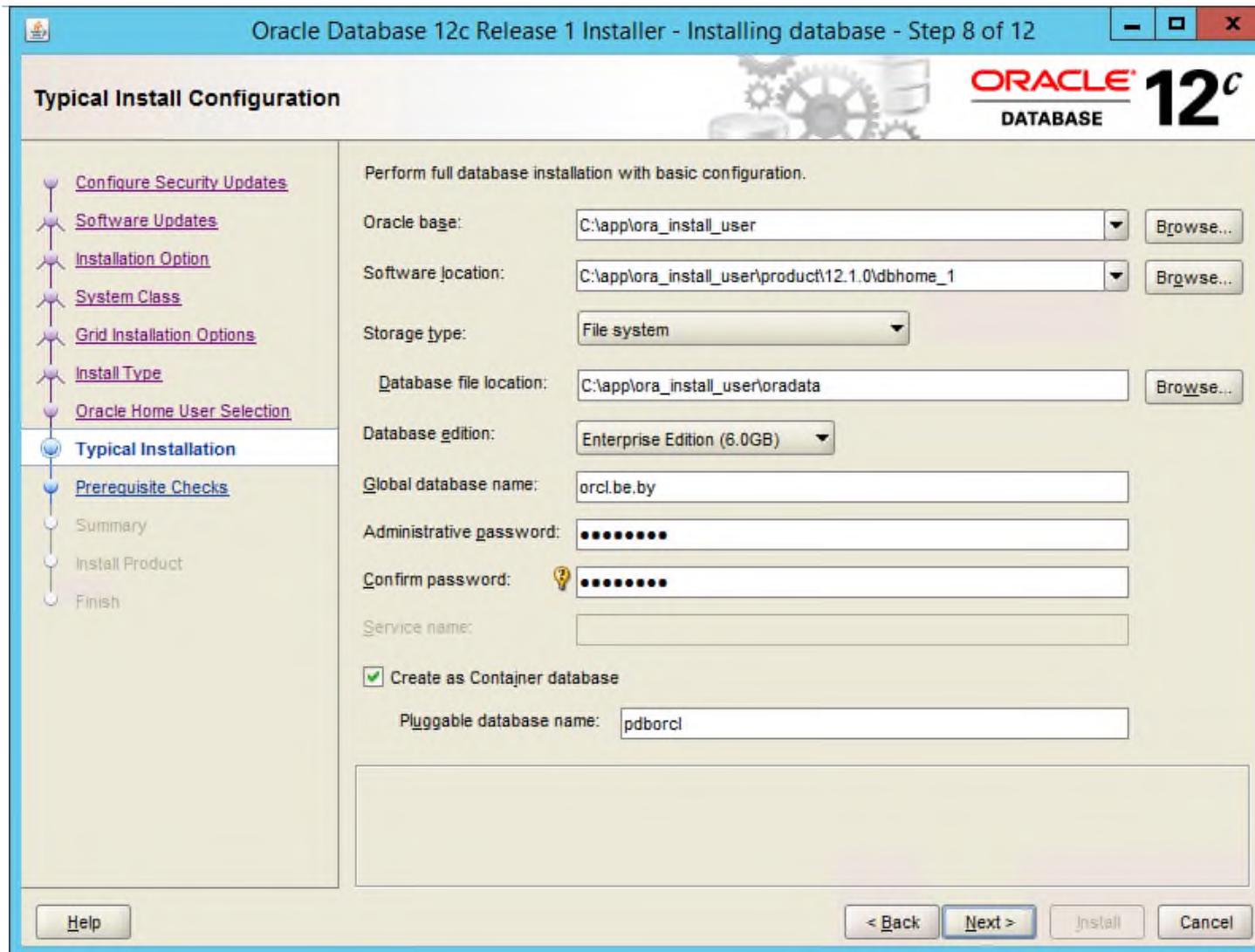
# Инсталляция ORACLE 12c EE



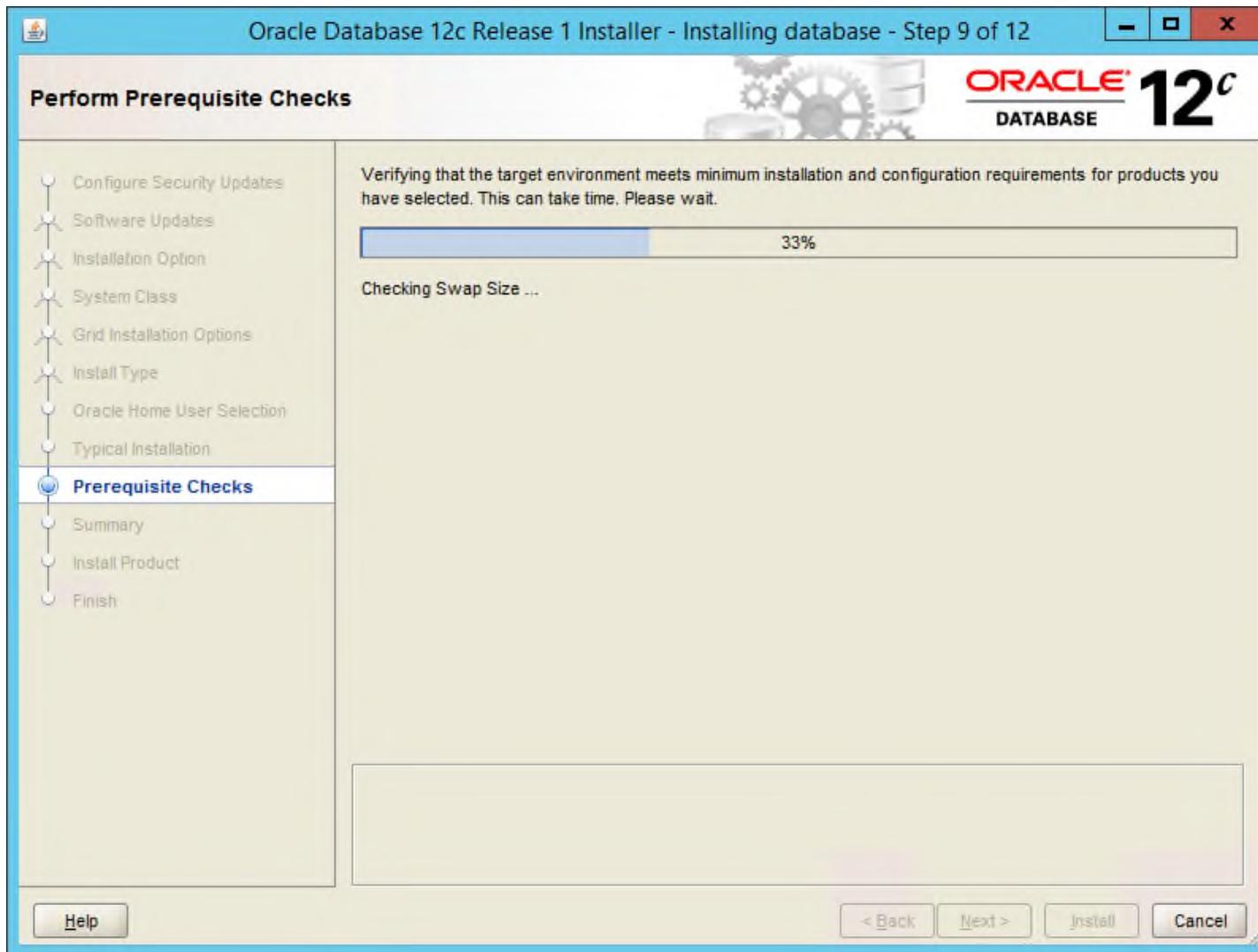
# Инсталляция ORACLE 12c EE



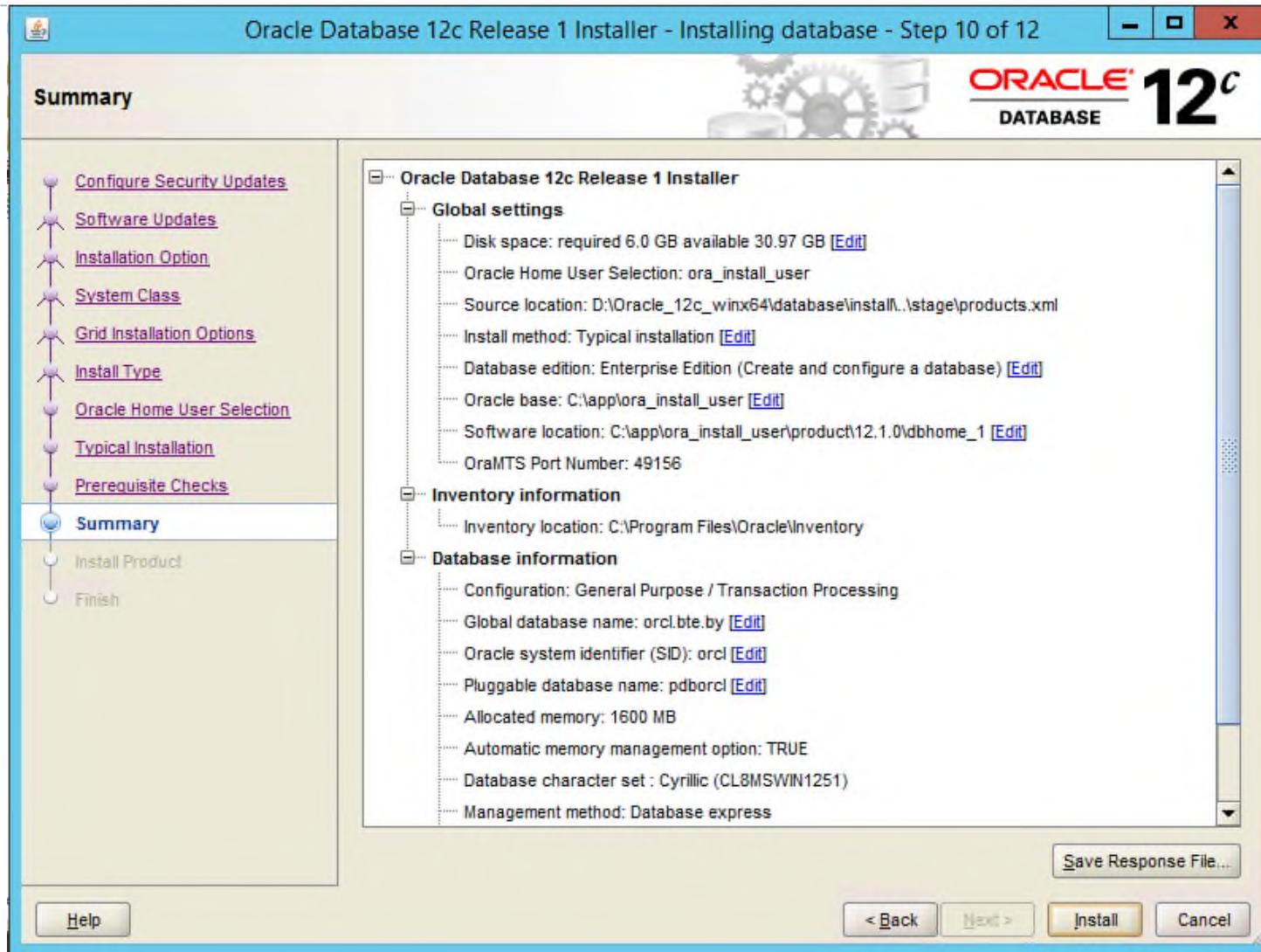
# Инсталляция ORACLE 12c EE



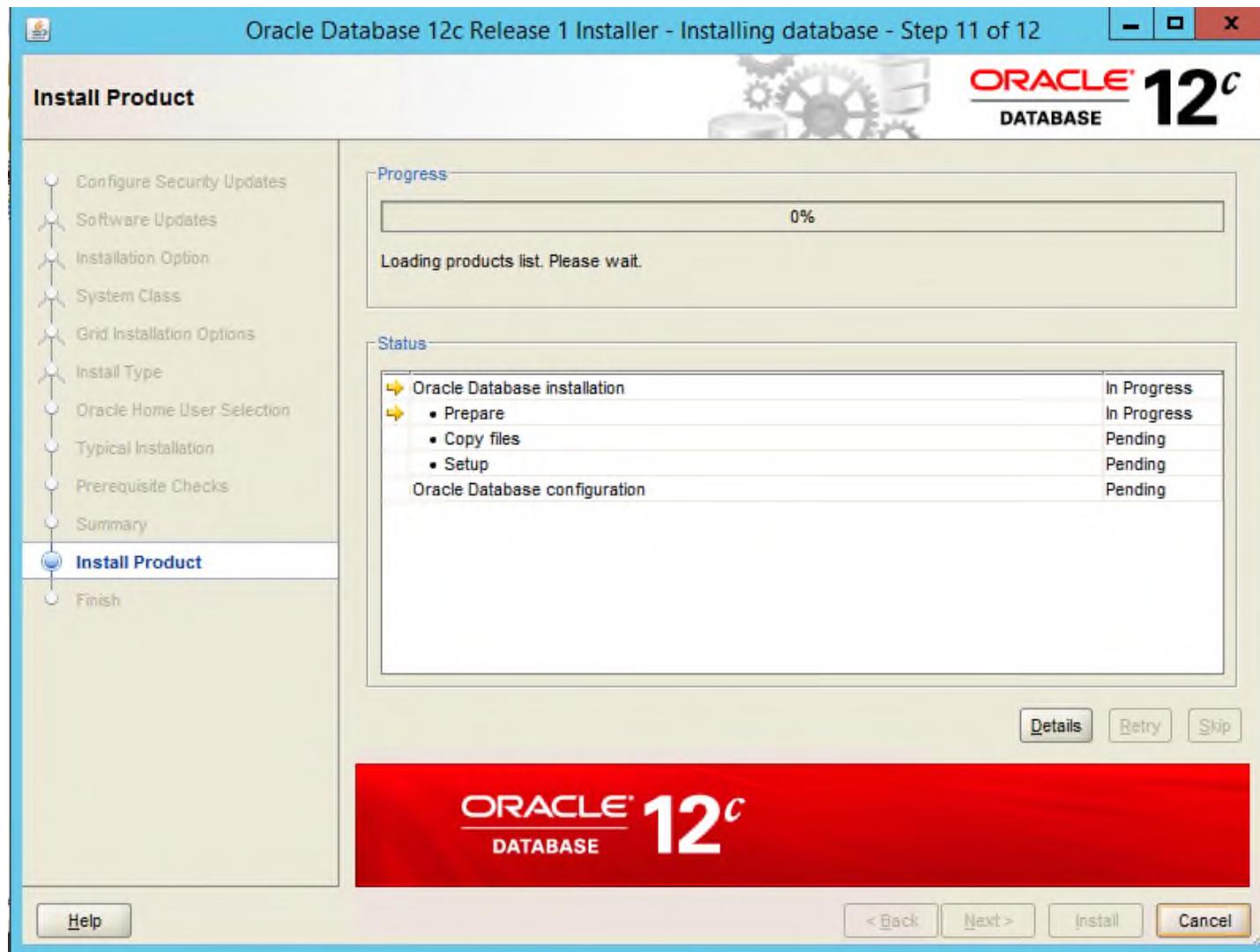
# Инсталляция ORACLE 12c EE



# Инсталляция ORACLE 12c EE



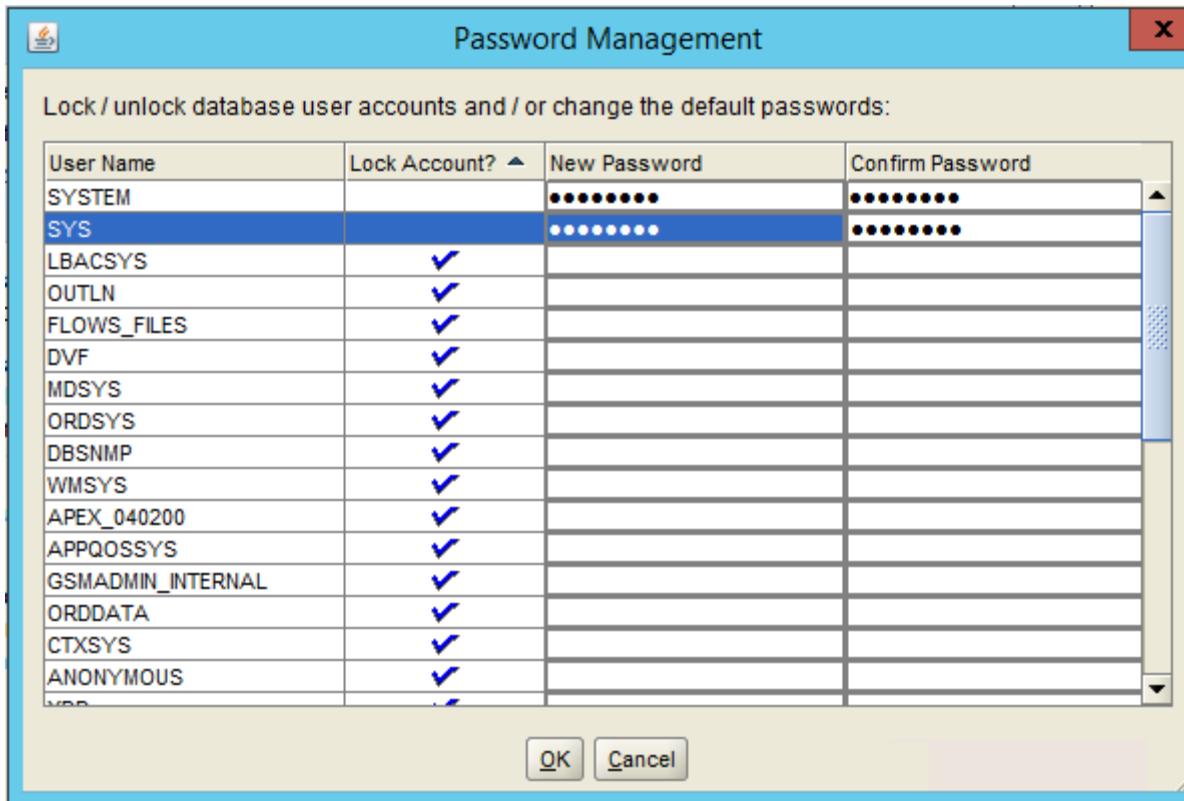
# Инсталляция ORACLE 12c EE



# Инсталляция ORACLE 12c EE



# Инсталляция ORACLE 12c EE



# Список пользователей и паролей по умолчанию

---

- ▶ SYSTEM – MANAGER
- ▶ SYS – CHANGE\_ON\_INSTALL
- ▶ SCOTT – TIGER



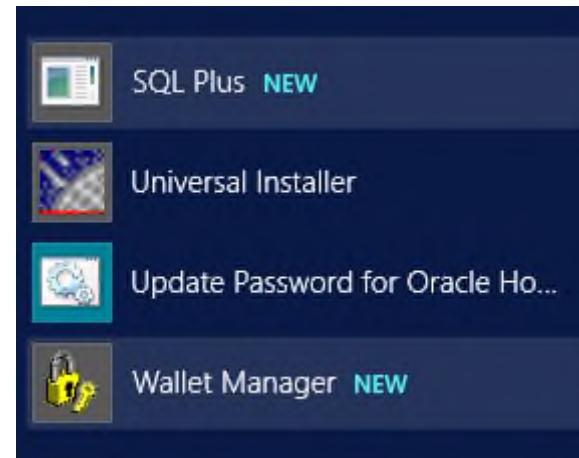
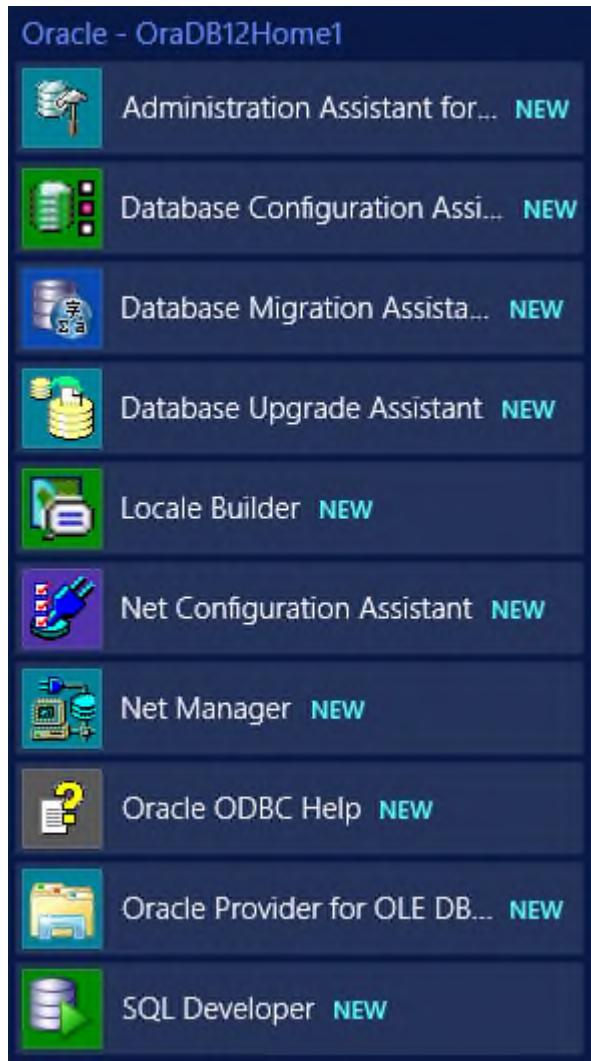
# Как проверить установку?

---

- ▶ Сервисы
- ▶ Список установленных программ
- ▶ Редактор реестра
- ▶ Установщик Universal Installer
- ▶ Пользователи и группы
- ▶ Соединение с базой данных



# Программы ORACLE 12c



# Сервисы ORACLE 12c

NAME	DESCRIPTION	STATUS	STARTUP TYPE	LOCATION
Network Service Interface Service	Helps the client...	Running	Automatic	Local System
Optimize drives	Helps the c...	Manual	Local System	
OracleJobSchedulerORCL		Disabled	Automatic	\ora_in\ora_job
OracleOraDB12Home1MTSRecoveryService		Running	Automatic	\ora_in\ora_mts
OracleOraDB12Home1TNSListener		Running	Automatic	\ora_in\ora_tns
OracleRemExecServiceV2		Running	Manual	Local System
OracleServiceORCL		Running	Automatic	\ora_in\ora_serv
OracleVssWriterORCL		Running	Automatic	\ora_in\ora_vss
Performance Counter DLL Host	Enables rem...	Manual	Local System	



# Пользователи и группы ORACLE 12c

 users	users are prevented from making ...
 ORA_ASMDBA	ORA_ASMDBA
 ORA_ASMOPER	ORA_ASMOPER
 ORA_CLIENT_LISTENERS	ORA_CLIENT_LISTENERS
 ora_dba	Oracle DBA Group
 ORA_GRID_LISTENERS	ORA_GRID_LISTENERS
 ORA_INSTALL	ORA_INSTALL
 ORA_OPER	ORA_OPER
 ORA_OraDB12Home1_DBA	ORA_OraDB12Home1_DBA
 ora_dba_1_oraop	ora_dba_1_oraop



# Редактор реестра

Редактор реестра

Файл Правка Вид Избранное Справка

Компьютер

- HKEY\_CLASSES\_ROOT
- HKEY\_CURRENT\_USER
- HKEY\_LOCAL\_MACHINE**
- BCD000000000
- HARDWARE
- SAM
- SECURITY
- SOFTWARE**
- ATI Technologies
- CBTEST
- Classes
- Clients
- Macromedia
- Microsoft
- ODBC
- ORACLE
- KEY\_OraDB12Home1**
- ODE
- OLEDB
- ODP.NET
- OradeMTSRecoveryServices
- Policies
- RegisteredApplications
- Wow6432Node

Имя	Тип	Значение
(По умолчанию)	REG_SZ	(значение не присвоено)
MSHELP_TOOLS	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS\MSHELP
NLS_LANG	REG_SZ	AMERICAN_AMERICA.WE8MSWIN1252
OLEDB	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS\oledb\mesg
OMTSRECO_PORT	REG_EXPAND_SZ	2030
ORA_ISUS_AUTOSTART	REG_EXPAND_SZ	TRUE
ORA_ISUS_SHUTDOWN	REG_EXPAND_SZ	TRUE
ORA_ISUS_SHUTDOWN_TIMEOUT	REG_EXPAND_SZ	90
ORA_ISUS_SHUTDOWNTYPE	REG_EXPAND_SZ	immediate
ORACLE_BASE	REG_SZ	C:\app\oracle
ORACLE_BUNDLE_NAME	REG_SZ	Enterprise
ORACLE_GROUP_NAME	REG_SZ	Oracle - OraDB12Home1
ORACLE_HOME	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS
ORACLE_HOME_KEY	REG_SZ	SOFTWARE\ORACLE\KEY_OraDB12Home1
ORACLE_HOME_NAME	REG_SZ	OraDB12Home1
ORACLE_HOME_TYPE	REG_SZ	1
ORACLE_SID	REG_SZ	ISUS
ORACLE_SVCUSER	REG_SZ	oracle
ORACLE_SVCUSER_PWDREQ	REG_SZ	1
RDBMS_ARCHIVE	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS\DATABASE...
RDBMS_CONTROL	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS\DATABASE
SQLPATH	REG_SZ	C:\app\oracle\product\12.1.0\dbhome_ISUS\ dbs

# Основные системные пользователи Oracle

---

- ▶ **SYS** – предопределенный привилегированный пользователь ранга администратора базы данных, который является владельцем ключевых ресурсов БД Oracle
- ▶ **SYSTEM** – предопределенный привилегированный пользователь, которому принадлежат ключевые ресурсы БД Oracle



# Системная роль Администратора базы данных

---

**DBA** – предопределенная роль, которая автоматически создаётся для каждой базы данных Oracle и содержит все системные привилегии, кроме SYSDBA и SYSOPER



# Специальные системные привилегии

---

- ▶ SYSDBA и SYSOPER - специальные привилегии администратора, которые позволяют выполнять базовые задачи администрирования: запуск или остановка экземпляра БД; создание, удаление, открытие или монтирования базы и др.
- ▶ Роль DBA не включает SYSDBA и SYSOPER. Привилегии могут быть указаны при подключении (connect) пользователя к БД



# Установка соединения с экземпляром

---

- ▶ SQL plus
- ▶ SQL Developer

## SQL Developer:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

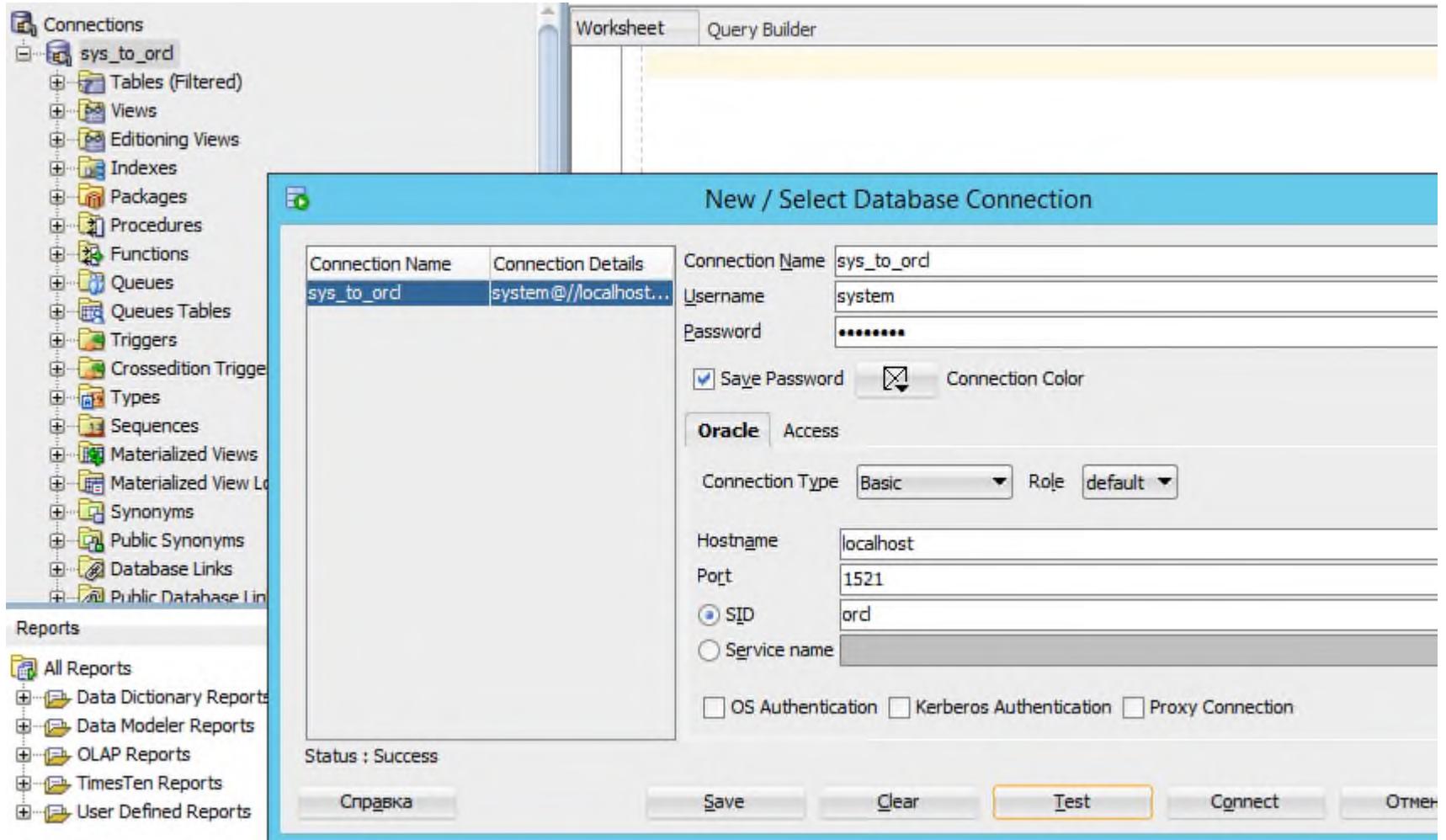


# Установка соединения SQLPlus

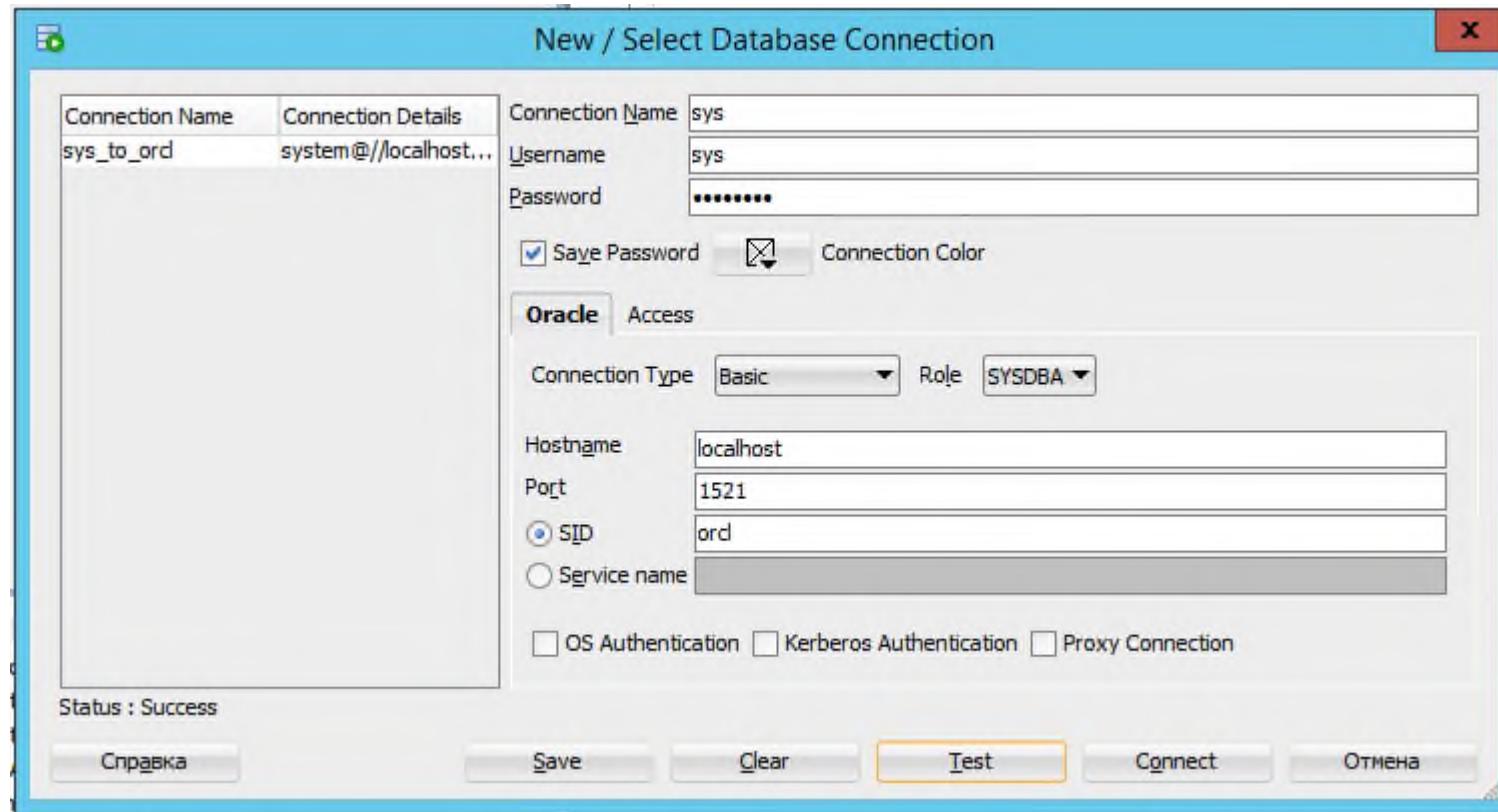
```
Administrator: C:\Windows\system32\cmd.exe - sqlplus /nolog  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
C:\Users\serve>sqlplus /nolog  
SQL*Plus: Release 12.1.0.1.0 Production on Mon Sep 5 04:49:38 2016  
Copyright (c) 1982, 2013, Oracle. All rights reserved.  
SQL> connect /as sysdba  
Connected.  
SQL> _
```

```
Administrator: C:\Windows\system32\cmd.exe - sqlplus /nolog  
Microsoft Windows [Version 6.3.9600]  
(c) 2013 Microsoft Corporation. All rights reserved.  
C:\Users\serve>sqlplus /nolog  
SQL*Plus: Release 12.1.0.1.0 Production on Mon Sep 5 04:49:38 2016  
Copyright (c) 1982, 2013, Oracle. All rights reserved.  
SQL> connect /as sysdba  
Connected.  
SQL> connect system/Pa$$w0rdP0rcl  
Connected.  
SQL> _
```

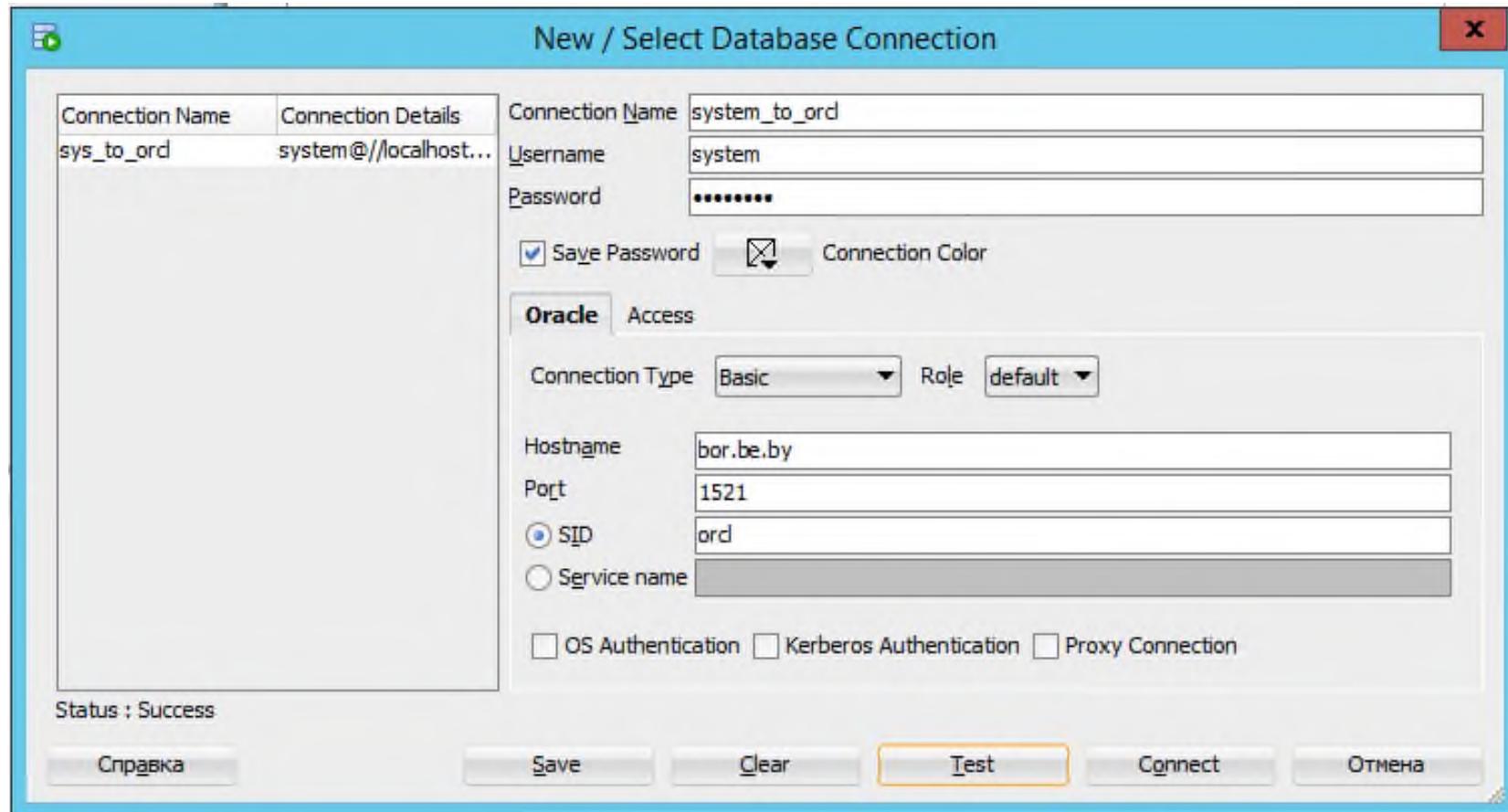
# Установка соединения SQL Developer



# Установка соединения SQL Developer



# Установка соединения SQL Developer



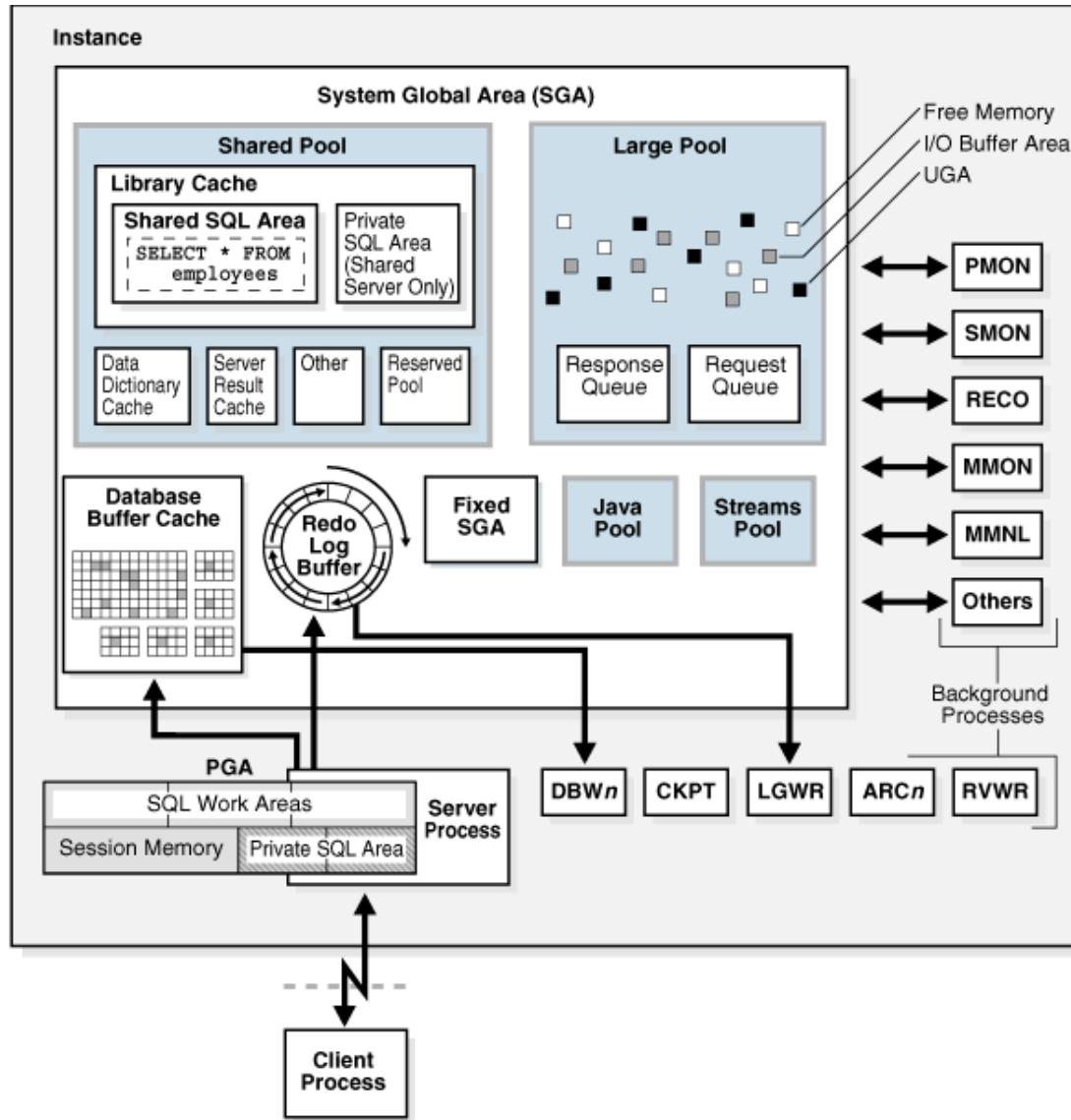
# Экземпляр Oracle (инстанс)

---

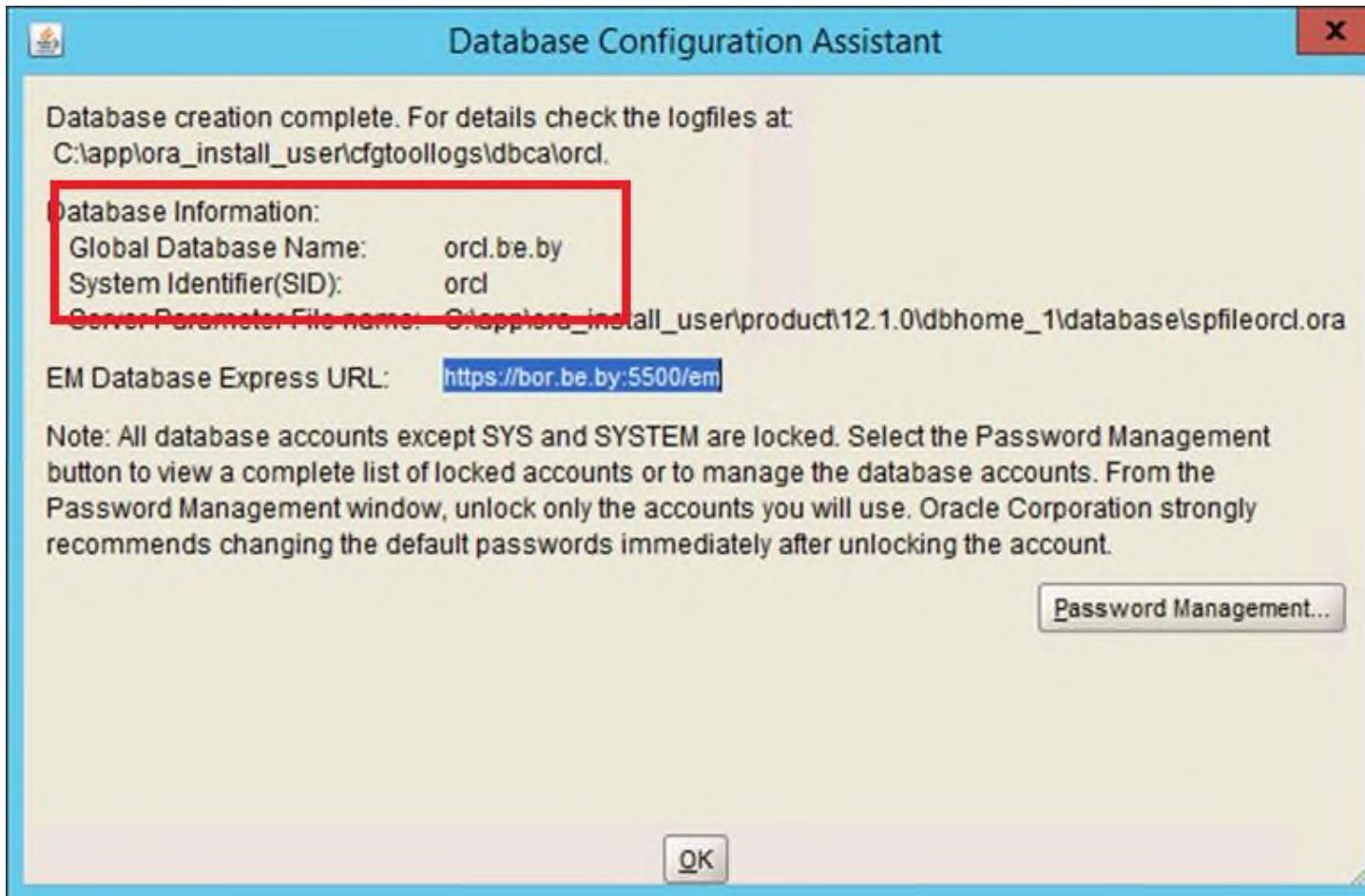
- ▶ запущенный сервер (программа) СУБД Oracle
- ▶ общая (глобальная) область памяти (SGA – *system global area*) и др. системные области памяти
- ▶ фоновые процессы, предназначенные для управления файлами базы данных



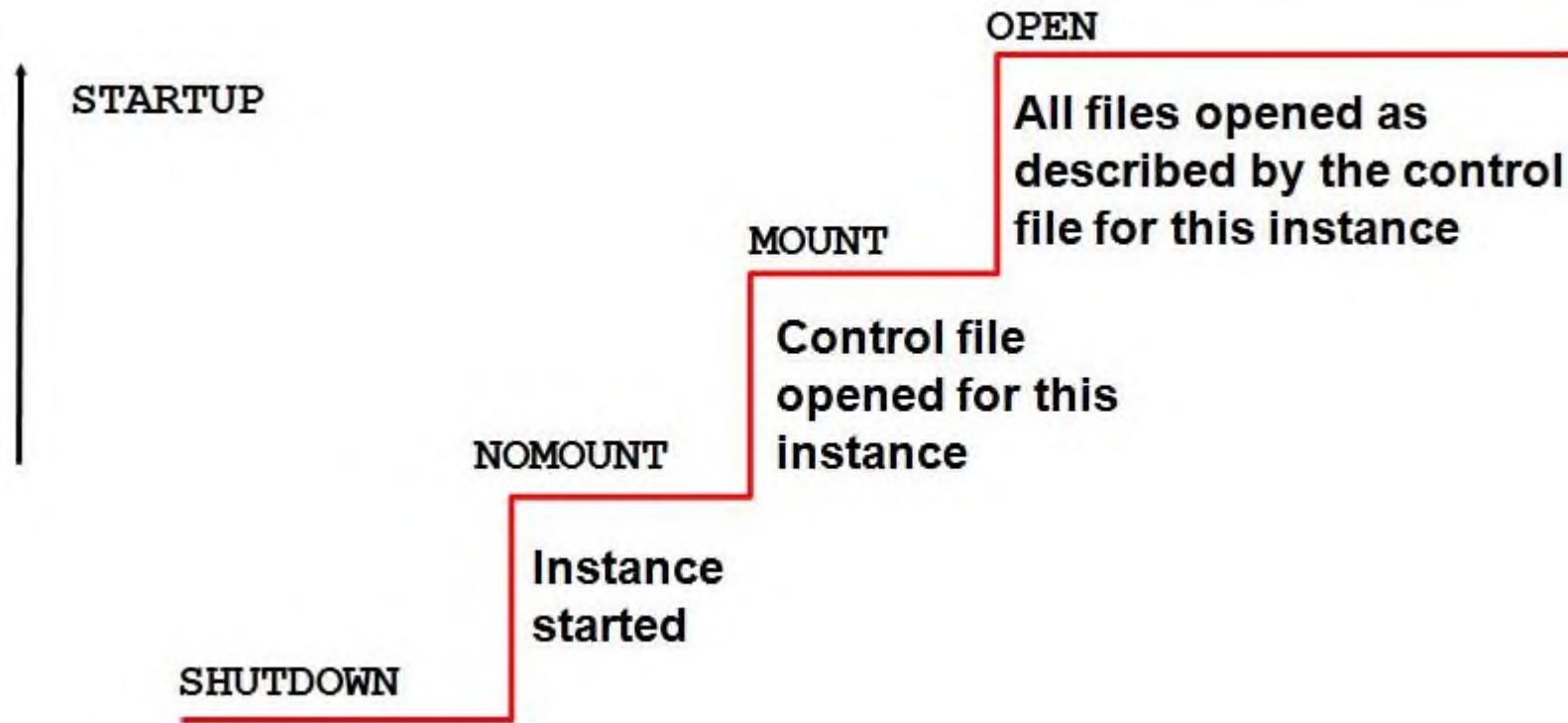
# Архитектура экземпляра Oracle



# SID – system identifier и Global Database Name



# Запуск и остановка экземпляра



# Запуск и остановка экземпляра

```
SQL> connect /as sysdba;
Connected.
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

```
SQL> connect /as sysdba;
Connected to an idle instance.
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> startup;
ORACLE instance started.

Total System Global Area 1720328192 bytes
Fixed Size          2403496 bytes
Variable Size       1023411032 bytes
Database Buffers   687865856 bytes
Redo Buffers        6647808 bytes
Database mounted.
Database opened.
SQL>
```



# Запуск и остановка экземпляра

```
0: Командная строка - sqlplus /nolog
SQL>
SQL>
SQL> connect /as sysdba;
Connected to an idle instance.
SQL>
SQL>
SQL>
SQL>
SQL> startup nomount;
ORACLE instance started.

Total System Global Area 1720328192 bytes
Fixed Size          2403496 bytes
Variable Size       1023411032 bytes
Database Buffers    687865856 bytes
Redo Buffers        6647808 bytes
SQL>
SQL>
SQL> alter database mount;
Database altered.

SQL> alter database open;
Database altered.

SQL>
```



# Вопросы?

---



# **Администрирование баз данных и приложений**

**Основные понятия и определения**

**Лекция 2**

# В этой лекции

---

- ▶ Словарь данных Oracle и представления словаря
- ▶ Подключаемые базы данных – pluggable / container databases
- ▶ Построение инфраструктуры базы данных



# Словарь Oracle

## Таблицы базы данных Oracle

- ▶ Таблицы пользователя

- Набор таблиц, созданных и обслуживаемых пользователем
- Содержат информацию пользователя

- ▶ Словарь данных

- Набор таблиц, созданных и обслуживаемых сервером Oracle
- Содержат информацию о базе данных



# Словарь Oracle

---

- ▶ Словарь Oracle - набор таблиц и связанных с ними представлений, который представляет возможность отследить внутреннюю структуру базы данных и деятельность СУБД Oracle



# Словарь данных

---

- ▶ Создается при генерации базы данных
- ▶ Обновляется и обслуживается сервером Oracle в фоновом режиме после выполнения операторов DDL
- ▶ Позволяет запрашивать данные в виде представлений
- ▶ Содержит следующую информацию:
  - Имена пользователей сервера Oracle
  - Уровни привилегий пользователей
  - Имена объектов базы данных
  - Табличные ограничения
  - Учетные данные



# Описание словаря данных

---

- ▶ Располагается в системном табличном пространстве **SYSTEM**
- ▶ Владелец: пользователь **SYS**, некоторые представления - **SYSTEM**
- ▶ Для доступа к словарю необходима специальная привилегия **GRANT SELECT ANY DICTIONARY**



# Запросы к словарю данных

---

- USER    Объекты, принадлежащие пользователю
  - ALL      Объекты, к которым пользователь имеет доступ
  - DBA     Все объекты базы данных (для администратора БД)
  - V\$       Производительность сервера
- ▶ Прочие представления (DICTIONARY и пр.)



# Примеры запросов к словарю данных

---

- ▶ Вывод списка всех представлений словаря данных, доступных пользователю

```
SELECT * FROM DICTIONARY;
```

- ▶ Вывод структуры представления USER\_OBJECTS

```
DESCRIBE user_objects;
```

- ▶ Вывод имен всех таблиц пользователя

```
SELECT object_name FROM user_objects WHERE object_type ='TABLE';
```



# Примеры запросов к словарю данных

---

- ▶ Просмотр типов объектов, принадлежащих пользователю, с помощью ключевого слова DISTINCT

```
SELECT DISTINCT object_type FROM user_objects;
```

- ▶ Просмотр подключений к серверу

```
SELECT * FROM v$session WHERE username is not null;
```

- ▶ Поиск таблиц словаря данных по заданным темам в столбце COMMENTS таблицы DICTIONARY

```
SELECT * FROM dictionary WHERE LOWER(comments) LIKE '%grant%';
```



# Примеры запросов к словарю данных

---

- ▶ Вывод списка объектов базы данных
  - ▶ `SELECT object_name, owner, status FROM dba_objects;`
- ▶ Вывод списка пользовательских констрайнов
  - ▶ `SELECT constraint_name, constraint_type  
FROM user_constraints WHERE table_name = '...';`
- ▶ Просмотр сведений о подключаемых БД
  - ▶ `SELECT name, open_mode, total_size FROM v$pdbs;`



# Словарь - заключение

---

- ▶ Словарь данных - это набор таблиц
- ▶ Пользователь просматривает содержимое словаря данных с помощью представлений
- ▶ Список основных представлений приводится в конце лекции



# Мультиарендная архитектура

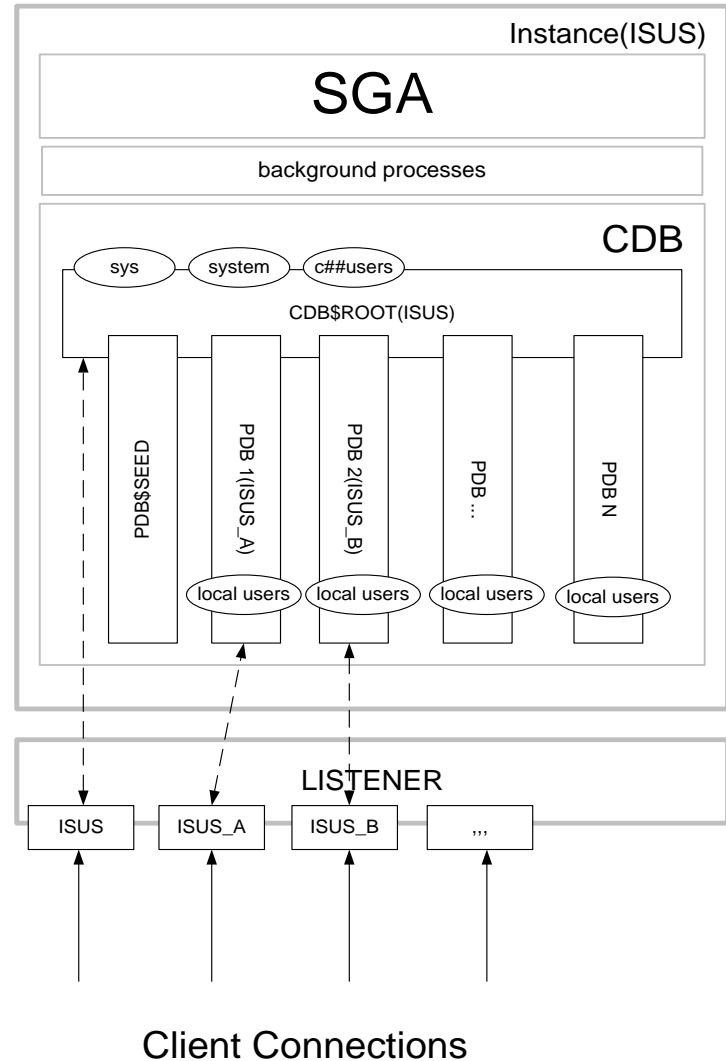
---

- ▶ Oracle Multitenant - технология, позволяющая запустить несколько независимых баз данных в рамках одного экземпляра.
- ▶ Каждая база данных имеет свой набор табличных пространств и набор схем, но при этом у них общая SGA и один набор серверных процессов.
- ▶ Базы данных изолированы, друг о друге ничего не знают, не конфликтуют между собой.
- ▶ Словарь разбивается на две части: общую часть и локальную.



# Мультиарендная архитектура

- ▶ CDB - container DB – контейнер базы данных
- ▶ PDB - pluggable DB – подключаемые базы данных



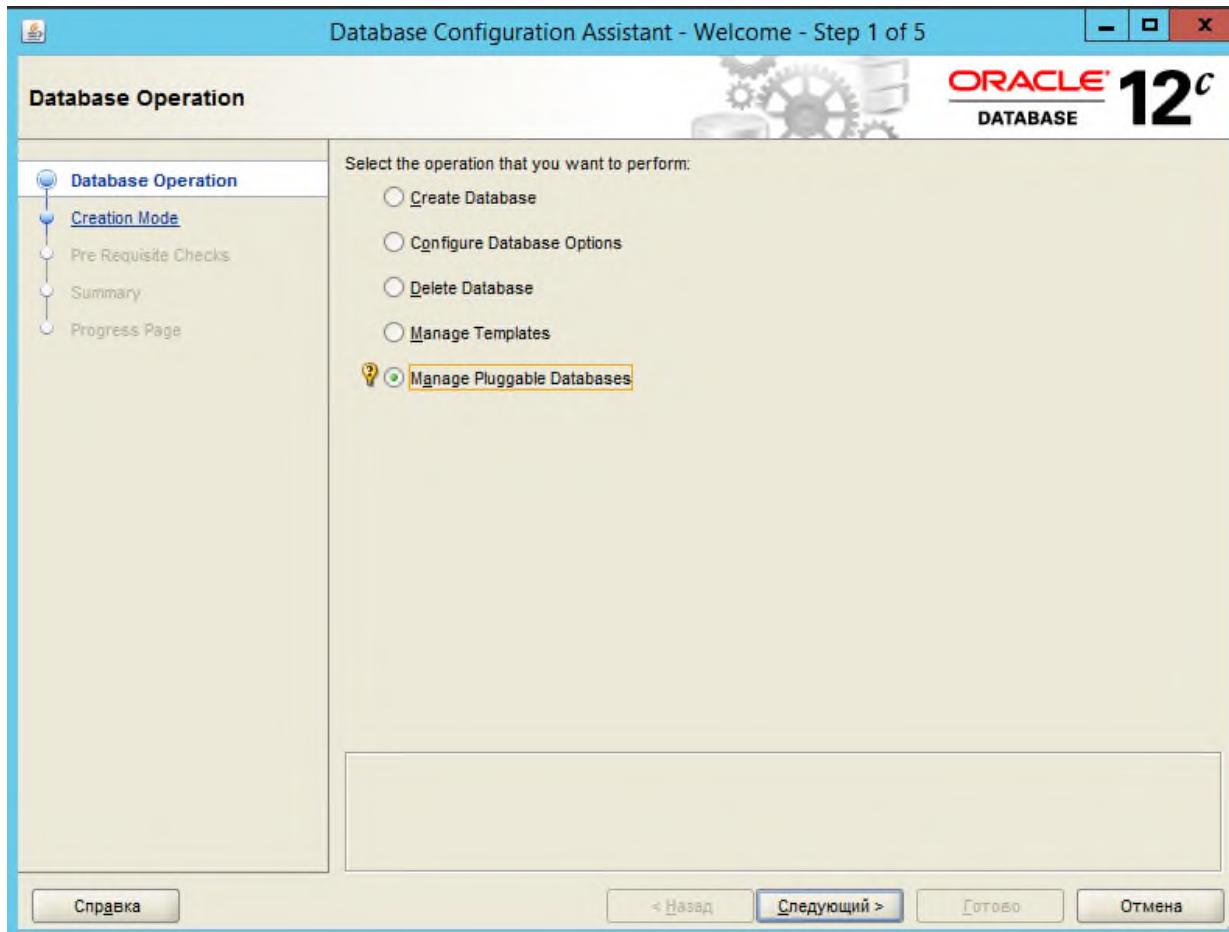
# Oracle Multitenant

---

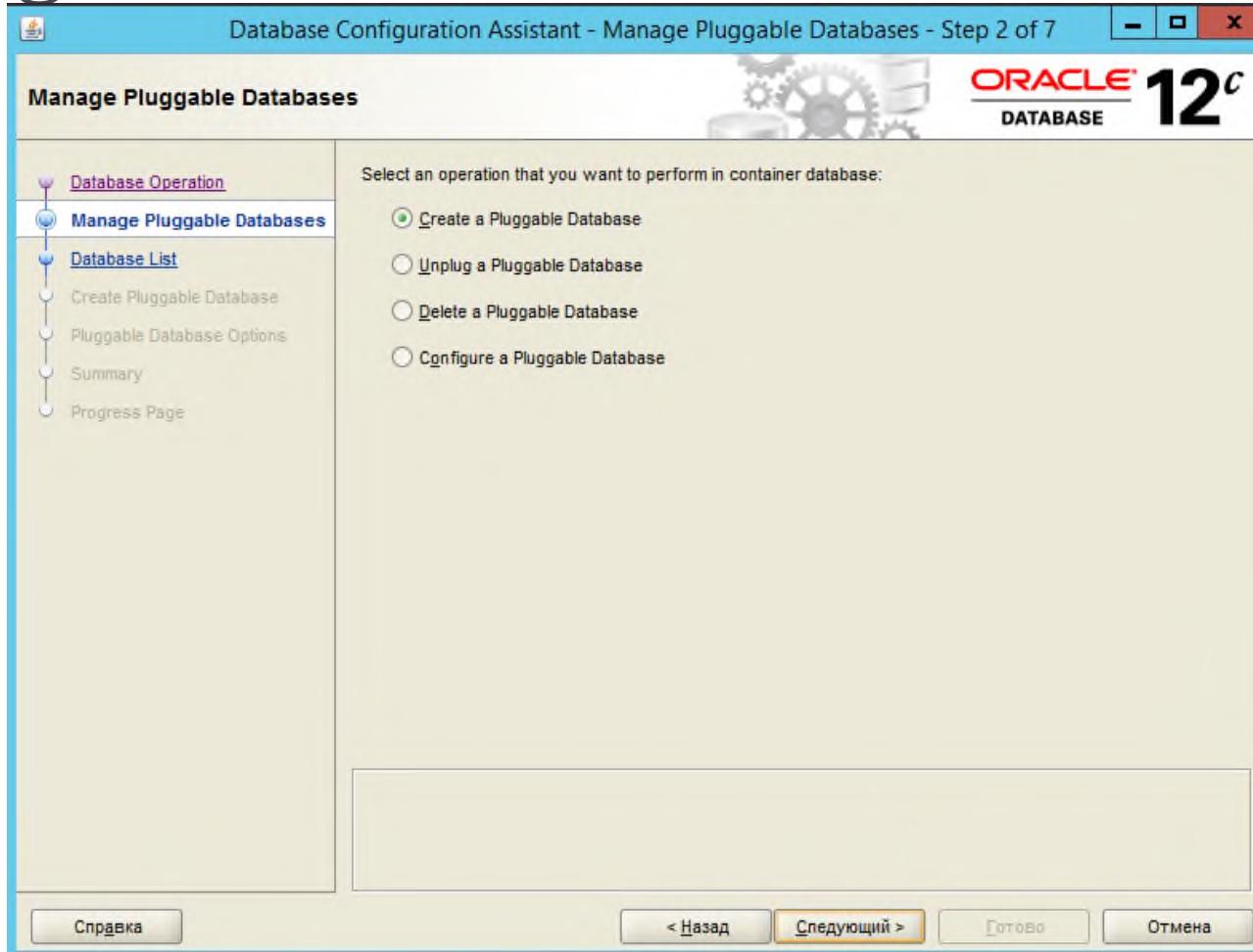
- ▶ Можно создавать несколько CDB – для разных версий программного обеспечения СУБД.
- ▶ Одну и ту же PDB можно переносить между CDB.
- ▶ В CDB создается главный контейнер Root. Root содержит метаданные CDB.
- ▶ В одной CDB можно создать до 252 PDB.



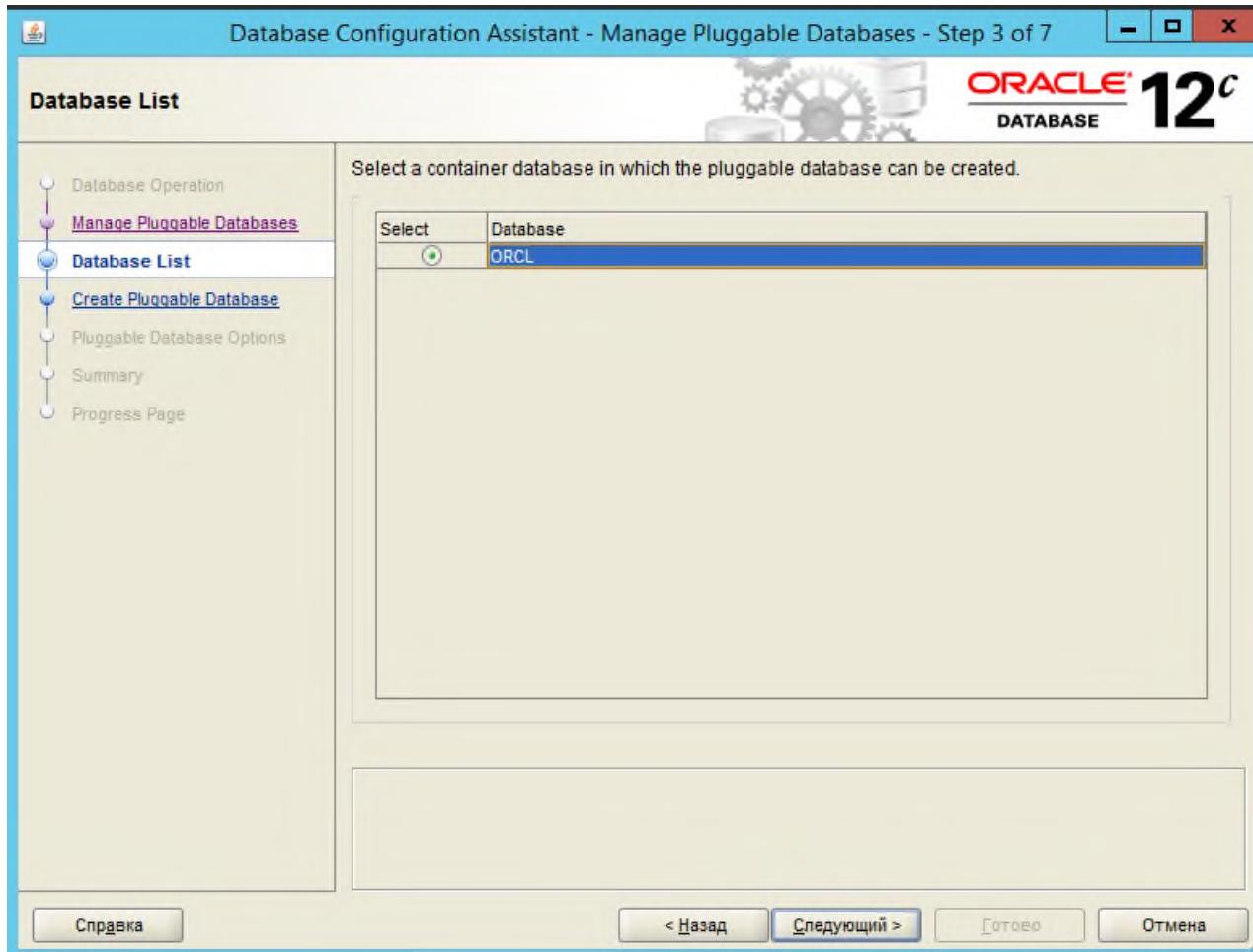
# Создание PDB в Oracle Database Configuration Assistant



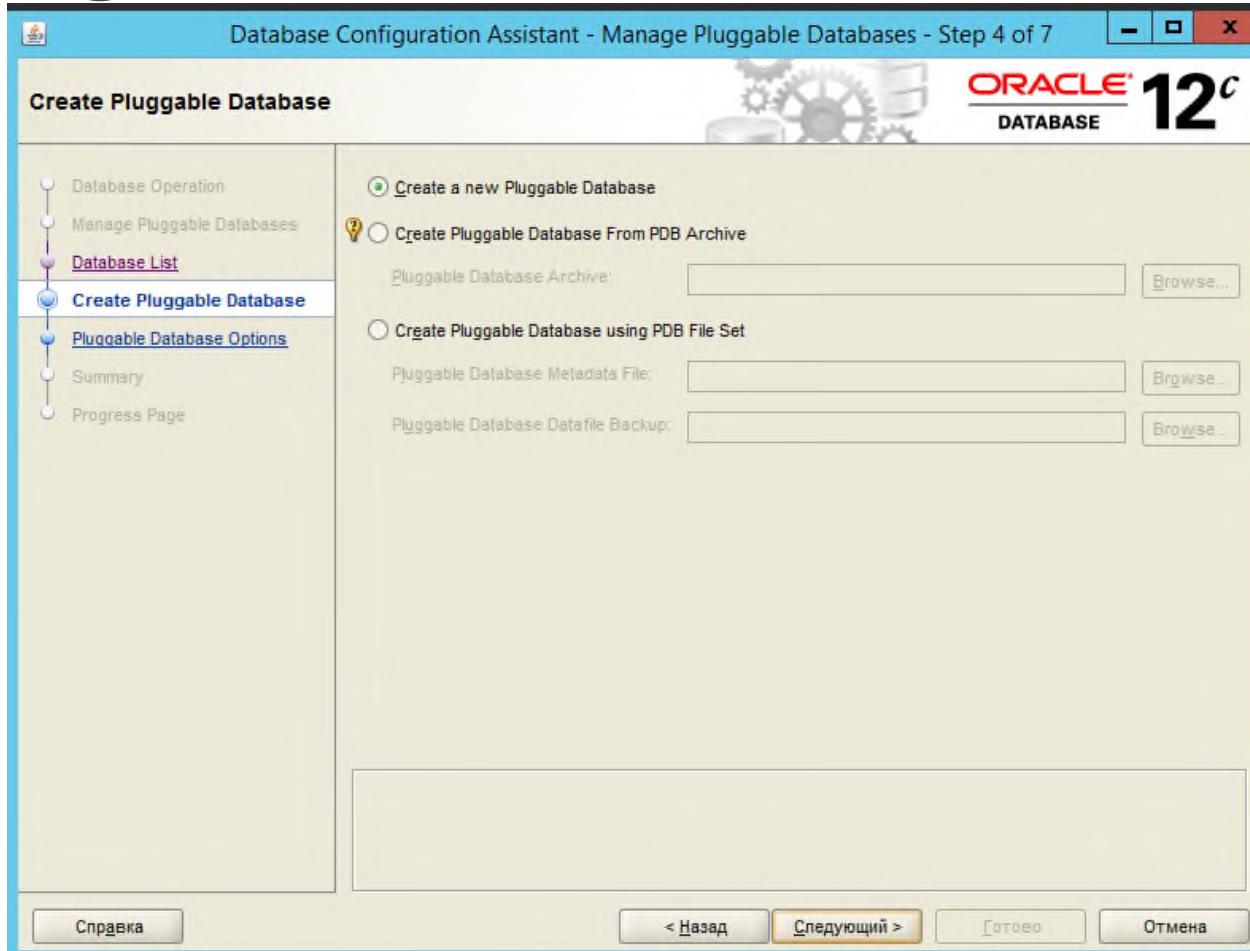
# Создание PDB в Oracle Database Configuration Assistant



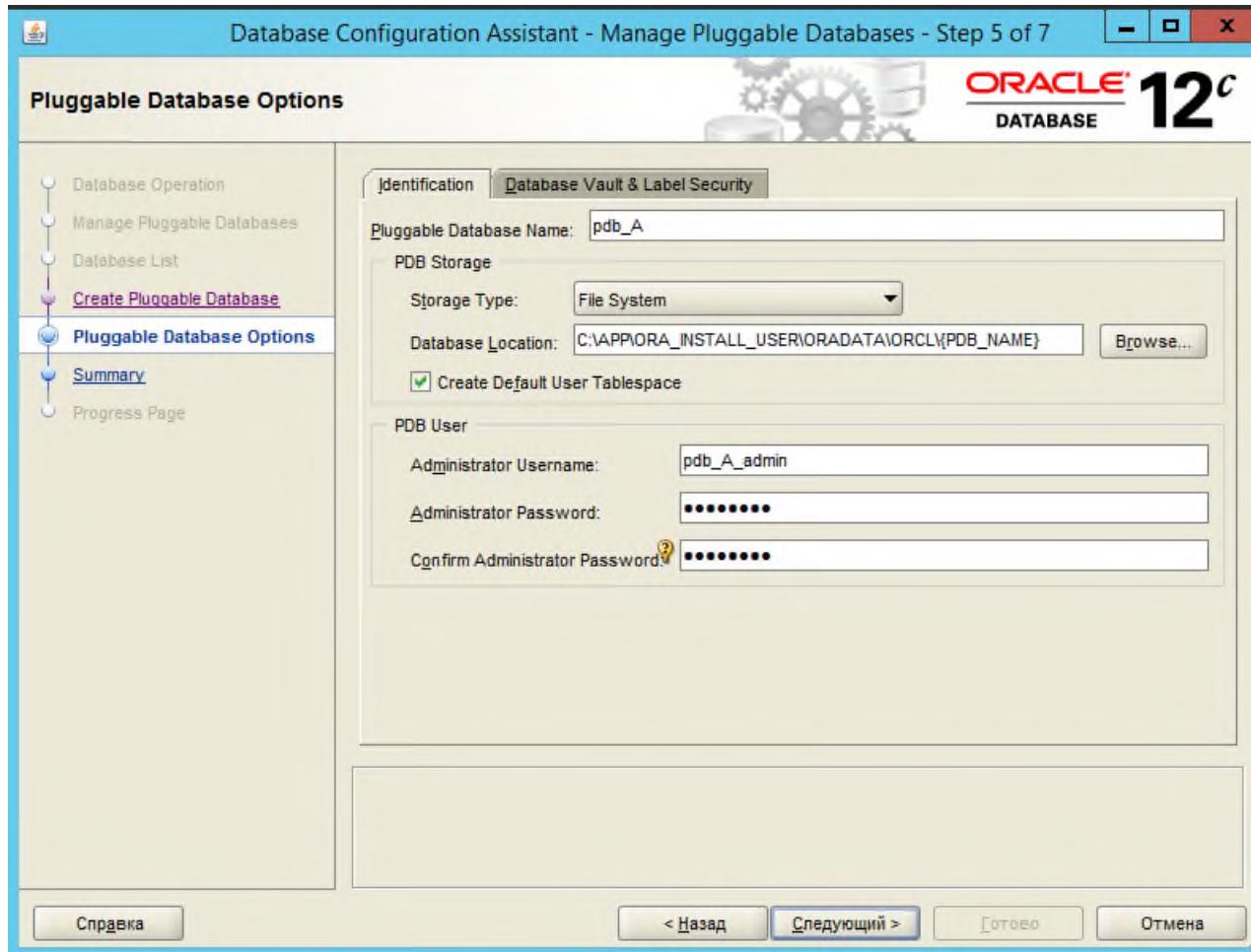
# Создание PDB в Oracle Database Configuration Assistant



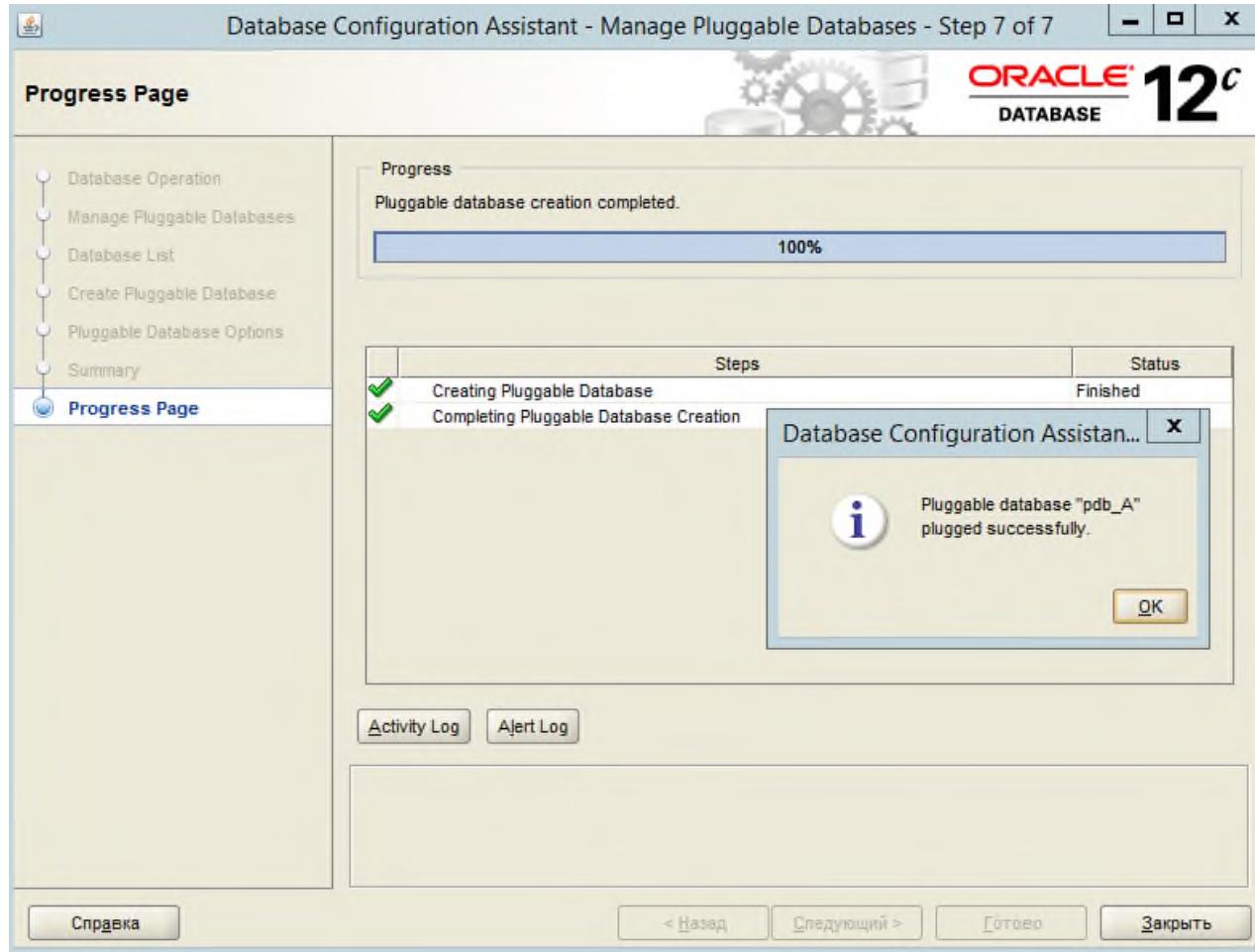
# Создание PDB в Oracle Database Configuration Assistant



# Создание PDB в Oracle Database Configuration Assistant



# Создание PDB в Oracle Database Configuration Assistant



# Соединение с PDB и действия над ней

```
SQL> connect /as sysdba
Connected.
SQL> select name, open_mode from v$pdbs;
NAME          OPEN_MODE
-----        -----
PDB$SEED      READ ONLY
PDBORCL       MOUNTED
PDB_A         READ WRITE
SQL> _
```

```
SQL> connect system/Pa$$w0rd@//localhost:1521/pdb_a.be.by as sysdba;
Connected.
SQL> _
```

```
SQL> alter pluggable database pdb_a unplug into 'C:\app\ora_install_user\pdb_a.xml';
Pluggable database altered.

SQL> drop pluggable database pdb_a;
Pluggable database dropped.

SQL> select name, open_mode from v$pdbs;
NAME          OPEN_MODE
-----        -----
PDB$SEED      READ ONLY
PDBORCL       MOUNTED
```

# Соединение с PDB и действия над ней

```
SQL> create pluggable database pdb_a as clone using 'C:\app\ora_install_user\pb_a.xml' nocopy tempfile reuse;
```

```
Pluggable database created.
```

```
SQL> alter pluggable database pdb_a open;
```

```
Pluggable database altered.
```

```
SQL> select name, open_mode from v$pdbs;
```

NAME	OPEN_MODE
PDB\$SEED	READ ONLY
PDBORCL	MOUNTED
PDB_A	READ WRITE

```
SQL> connect pdb_a_admin/Pa$$w0rd@//localhost:1521/pdb_a.be.by as sysdba;  
Connected.
```

```
SQL> select name, open_mode from v$pdbs;
```

NAME	OPEN_MODE
PDB_A	READ WRITE

# Создание общих пользователей

```
SQL> create user c##cdb_admin identified by Pa$$w0rd;
User created.

SQL> connect c##cdb_admin/Pa$$w0rd;
ERROR:
ORA-01045: user C##CDB_ADMIN lacks CREATE SESSION privilege; logon denied

Warning: You are no longer connected to ORACLE.
SQL> connect/ as sysdba;
Connected.
SQL> grant create session to c##cdb_admin;

Grant succeeded.

SQL> connect c##cdb_admin/Pa$$w0rd;
Connected.
SQL> connect c##cdb_admin/Pa$$w0rd@//localhost:1521/pdb_a.be.by;
ERROR:
ORA-01045: user C##CDB_ADMIN lacks CREATE SESSION privilege; logon denied

Warning: You are no longer connected to ORACLE.
SQL> connect pdb_a_admin/Pa$$w0rd@//localhost:1521/pdb_a.be.by as sysdba;
Connected.
SQL> grant create session to c##cdb_admin;

Grant succeeded.

SQL> connect c##cdb_admin/Pa$$w0rd@//localhost:1521/pdb_a.be.by;
Connected.
SQL> _
```



# Подключение в SQL Developer

New / Select Database Connection

Connection Name	Connection Details
pdb_a_admin_to_pdb_a	pdb_a_admin@//localhost:1521/pdb_a
pdb_b_admin_to_pdb_b	pdb_b_admin@//localhost:1521/pdb_b
sys	sys@//localhost:1521
sys_to_ord	sys@//localhost:1521/orcl
system_to_ord	system@//localhost:1521/orcl

Connection Name: **pdb\_a\_admin\_to\_pdb\_a**  
Username: **pdb\_a\_admin**  
Password: **\*\*\*\*\***  
 Save Password  Connection Color

**Oracle** Access

Connection Type: **Basic** Role: **default**

Hostname: **localhost**  
Port: **1521**  
 SID  
 Service name: **pdb\_a.be.by**

OS Authentication  Kerberos Authentication  Proxy Connection

Status : Success

**Справка** **Save** **Clear** **Test** **Connect** **От**

# Создание инфраструктуры базы данных

---

- ▶ создание табличных пространств
- ▶ создание ролей
- ▶ назначение ролям системных привилегий
- ▶ создание профилей безопасности
- ▶ создание пользователей
- ▶ назначение пользователям ролей
- ▶ создание объектов базы данных
- ▶ назначение ролям объектных привилегий



# Табличные пространства - TABLESPACES

---

- ▶ Табличное пространство – логическая структура хранения данных, контейнер сегментов
- ▶ С одним табличным пространством связаны один или несколько файлов операционной системы, с каждым файлом связано только одно табличное пространство



# Табличные пространства - TABLESPACES

---

- ▶ PERMANENT - хранение постоянных объектов БД
- ▶ TEMPORARY - хранение временных данных
- ▶ UNDO - хранение сегментов отката, используется всегда один



# TABLESPACES

---

- ▶ PERMANENT-пространств может быть несколько
- ▶ можно создавать свои объекты в любом из PERMANENT-пространств, если это не запрещено,
- ▶ одно может приписано в качестве пространства по умолчанию.
- ▶ TEMPORARY-пространств может быть несколько
- ▶ TEMPORARY-пространство приписывается одному или нескольким пользователям, которые могут там размещать свои временные данные
- ▶ UNDO-пространств может быть несколько, но активным является всегда только одно (указывается в конфигурационном файле SPIFILE.ORA)



# PERMANENT TABLESPACES

---

```
CREATE TABLESPACE ts_ea
  DATAFILE 'C:\app\Tablespaces\ts_EA.dbf'
  SIZE 10 m
  AUTOEXTEND ON NEXT 500K
  MAXSIZE 100M
  EXTENT MANAGEMENT LOCAL;
```

```
DROP TABLESPACE TS_EA;
```



# PERMANENT TABLESPACES

```
select TABLESPACE_NAME, STATUS, contents logging from SYS.DBA_TABLESPACES;
```

	TABLESPACE_NAME	STATUS	LOGGING	
1	SYSTEM	ONLINE	PERMANENT	
2	SYSAUX	ONLINE	PERMANENT	
3	UNDOTBS1	ONLINE	UNDO	
4	TEMP	ONLINE	TEMPORARY	
5	USERS	ONLINE	PERMANENT	
6	MYTABLE_SPACE	ONLINE	PERMANENT	
7	MYTABLE_SPACE_TEMP	ONLINE	TEMPORARY	
8	TS_EA	ONLINE	PERMANENT	

# TEMPORARY TABLESPACES

```
CREATE TEMPORARY TABLESPACE ts_TEMP_ea
  TEMPFILE 'C:\app\Tablespaces\ts_TEMP_EA.dbf'
  SIZE 10 m
  AUTOEXTEND ON NEXT 500K
  MAXSIZE 100M
  EXTENT MANAGEMENT LOCAL;
```

TABLESPACE_NAME	STATUS	LOGGING
SYSTEM	ONLINE	PERMANENT
SYSAUX	ONLINE	PERMANENT
UNDOTBS1	ONLINE	UNDO
TEMP	ONLINE	TEMPORARY
USERS	ONLINE	PERMANENT
MYTABLE_SPACE	ONLINE	PERMANENT
MYTABLE_SPACE_TEMP	ONLINE	TEMPORARY
TS_EA	ONLINE	PERMANENT
TS_TEMP_EA	ONLINE	TEMPORARY

# TEMPORARY TABLESPACES

```
SELECT FILE_NAME, TABLESPACE_NAME, STATUS, MAXBYTES, USER_BYTES FROM DBA_DATA_FILES  
UNION  
SELECT FILE_NAME, TABLESPACE_NAME, STATUS, MAXBYTES, USER_BYTES FROM DBA_TEMP_FILES;
```

FILE_NAME	TABLESPACE_NAME	STATUS	MAXBYTES	USER_BYTES
1 C:\APP\ORA_INSTALL_USER\ORADATA\ORC...	SYSAUX	AVAILABLE	34359721984	932184064
2 C:\APP\ORA_INSTALL_USER\ORADATA\ORC...	SYSTEM	AVAILABLE	34359721984	816840704
3 C:\APP\ORA_INSTALL_USER\ORADATA\ORC...	TEMP	ONLINE	34359721984	66060288
4 C:\APP\ORA_INSTALL_USER\ORADATA\ORC...	UNDOTBS1	AVAILABLE	34359721984	759169024
5 C:\APP\ORA_INSTALL_USER\ORADATA\ORC...	USERS	AVAILABLE	34359721984	4194304
6 C:\APP\TABLESPACES\TS_EA.DBF	TS_EA	AVAILABLE	104857600	9437184
7 C:\APP\TABLESPACES\TS_LEC.DBF	MYTABLE_SPACE	AVAILABLE	2147483648	9437184
8 C:\APP\TABLESPACES\TS_LEC_TEMP.DBF	MYTABLE_SPACE_TEMP	ONLINE	2147483648	9437184
9 C:\APP\TABLESPACES\TS_TEMP_EA.DBF	TS_TEMP_EA	ONLINE	104857600	9437184

# Привилегии

---

- ▶ Привилегия - это право выполнять конкретный тип предложений SQL, или право доступа к объекту другого пользователя.
- ▶ ORACLE имеет два вида привилегий: системные и объектные.
- ▶ Назначаются оператором GRANT
- ▶ Отзываются оператором REVOKE



# Группы системных привилегий - примеры

Группа	Примеры привилегий
PROCEDURE	CREATE
PROFILE	CREATE ANY
ROLE	ALTER
ROLLBACK SEGMENT	ALTER ANY
SESSION	DROP
SEQUENCE	
SYSTEM	
TABLE	
TABLESPACE	
TRIGGER	
USER	
VIEW	



# Роли - ROLES

- ▶ Роль – это именованный набор привилегий

```
CREATE ROLE RLEACORE;
```

```
SELECT * FROM dba_roles WHERE ROLE LIKE 'RL%';
```

	ROLE	PASSWORD_REQUIRED	AUTHENTICATION_TYPE	COMMON	ORACLE_MAINTAINED
1	RLEACORE	NO	NONE	NO	N

```
GRANT CREATE SESSION,  
      CREATE TABLE,  
      CREATE VIEW,  
      CREATE PROCEDURE TO RLEACORE;
```

```
SELECT * FROM DBA_SYS_PRIVS WHERE GRANTEE = 'RLEACORE';
```

	GRANTEE	PRIVILEGE	ADMIN_OPTION	COMMON
1	RLEACORE	CREATE SESSION	NO	NO
2	RLEACORE	CREATE TABLE	NO	NO
3	RLEACORE	CREATE PROCEDURE	NO	NO
4	RLEACORE	CREATE VIEW	NO	NO

# Создание профиля безопасности

```
CREATE PROFILE PFEACORE LIMIT  
  PASSWORD_LIFE_TIME 180 -- количество дней жизни пароля  
  SESSIONS_PER_USER 3 -- количество сессий для пользователя  
  FAILED_LOGIN_ATTEMPTS 7 -- количество попыток входа  
  PASSWORD_LOCK_TIME 1 -- количество дней блокирования после ошибок  
  PASSWORD_REUSE_TIME 10 -- через сколько дней можно повторить пароль  
  PASSWORD_GRACE_TIME DEFAULT -- количество дней предупреждений о смене пароля  
  CONNECT_TIME 180 -- время соединения, минут  
  IDLE_TIME 30 -- количество минут простоя
```

# Свойства созданного профиля

```
SELECT * FROM DBA_PROFILES WHERE PROFILE = 'PFEACORE';
```

#	PROFILE	RESOURCE_NAME	RESOURCE_TYPE	LIMIT	COMMON	
1	PFEACORE	COMPOSITE_LIMIT	KERNEL	DEFAULT	NO	
2	PFEACORE	SESSIONS_PER_USER	KERNEL	3	NO	
3	PFEACORE	CPU_PER_SESSION	KERNEL	DEFAULT	NO	
4	PFEACORE	CPU_PER_CALL	KERNEL	DEFAULT	NO	
5	PFEACORE	LOGICAL_READS_PER_SESSION	KERNEL	DEFAULT	NO	
6	PFEACORE	LOGICAL_READS_PER_CALL	KERNEL	DEFAULT	NO	
7	PFEACORE	IDLE_TIME	KERNEL	30	NO	
8	PFEACORE	CONNECT_TIME	KERNEL	180	NO	
9	PFEACORE	PRIVATE_SGA	KERNEL	DEFAULT	NO	
10	PFEACORE	FAILED_LOGIN_ATTEMPTS	PASSWORD	7	NO	
11	PFEACORE	PASSWORD_LIFE_TIME	PASSWORD	180	NO	
12	PFEACORE	PASSWORD_REUSE_TIME	PASSWORD	10	NO	
13	PFEACORE	PASSWORD_REUSE_MAX	PASSWORD	DEFAULT	NO	

# Свойства профиля по умолчанию

```
SELECT * FROM DBA_PROFILES WHERE PROFILE = 'DEFAULT';
```

PROFILE	RESOURCE_NAME	RESOURCE_TYPE	LIMIT	COMMON
1 DEFAULT	COMPOSITE_LIMIT	KERNEL	UNLIMITED	NO
2 DEFAULT	SESSIONS_PER_USER	KERNEL	UNLIMITED	NO
3 DEFAULT	CPU_PER_SESSION	KERNEL	UNLIMITED	NO
4 DEFAULT	CPU_PER_CALL	KERNEL	UNLIMITED	NO
5 DEFAULT	LOGICAL_READS_PER_SESSION	KERNEL	UNLIMITED	NO
6 DEFAULT	LOGICAL_READS_PER_CALL	KERNEL	UNLIMITED	NO
7 DEFAULT	IDLE_TIME	KERNEL	UNLIMITED	NO
8 DEFAULT	CONNECT_TIME	KERNEL	UNLIMITED	NO
9 DEFAULT	PRIVATE_SGA	KERNEL	UNLIMITED	NO
10 DEFAULT	FAILED_LOGIN_ATTEMPTS	PASSWORD	10	NO
11 DEFAULT	PASSWORD_LIFE_TIME	PASSWORD	180	NO
12 DEFAULT	PASSWORD_REUSE_TIME	PASSWORD	UNLIMITED	NO
13 DEFAULT	PASSWORD_REUSE_MAX	PASSWORD	UNLIMITED	NO

# Создание пользователя

```
CREATE USER EACORE IDENTIFIED BY 12345  
DEFAULT TABLESPACE TBLSP_EA QUOTA UNLIMITED ON TBLSP_EA  
TEMPORARY TABLESPACE TBLSP_TEMP_EA  
PROFILE PFEACORE  
ACCOUNT UNLOCK  
PASSWORD EXPIRE
```

```
GRANT RLEACORE TO EACORE;
```



# Попытка подключения к БД

New / Select Database Connection

Connection Name	Connection Details
pdb_a_admin_to_pd...	pdb_a_admin@//loca...
pdb_b_admin_to_pd...	pdb_b_admin@//loca...
sys	sys@//localhost:152...
sys_to_ord	sys@//localhost:152...
system_to_ord	system@//localhost:...
system_to_pdb_a	system@//localhost:...

Connection Name: eacore  
Username: eacore  
Password: \*\*\*\*\*  
 Save Password  Connection Color

**Oracle** Access

Connection Type: Basic Role: default

Hostname: localhost  
Port: 1521  
 SID: xe  
 Service name: pdb\_a.be.by

OS Authentication  Kerberos Authentication  Proxy Connection

Status : Failure -Test failed: ORA-28001: the password has expired

Справка  Save  Clear  Test  Connect

# Попытка подключения к БД

```
SQL Plus

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Tue Sep 13 2016 00:36:31 -07:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

SQL> connect /as sysdba;
Connected.
SQL> connect eacore/12345@//localhost:1521/pdb_a.be.by
ERROR:
ORA-28001: the password has expired

Changing password for eacore
New password:
Retype new password:
Password changed
Connected.
SQL>
```

# Приложение – Некоторые представления словаря данных

ПРЕДСТАВЛЕНИЕ БАЗЫ ДАННЫХ	Описание
DBA_USERS	Хранит информацию о всех, кто имеет учетную запись в базе данных Oracle. Вместе с именем и хешированным паролем пользователя хранится имя назначенного ему пользователя.
DBA_PROFILE	Для каждого профиля хранит информацию о ресурсах и их лимитах.
DBA_ROLES	Детализирует все роли, содержащиеся в базе данных.
DBA_ROLE_PRIVS	Роли, которые были назначены конкретным пользователям и другим ролям.
DBA_SYS_PRIVS	Системные привилегии, которые были выданы конкретным пользователям или ролям.
DBA_TAB_PRIVS	Привилегии Select, Insert и Update, которые были выданы конкретным пользователям или ролям.



# Приложение – Некоторые представления словаря данных

ПРЕДСТАВЛЕНИЕ БАЗЫ ДАННЫХ	Описание
DBA_COL_PRIVS	Привилегии Select, Insert и Update, которые были выданы конкретным пользователям или ролям.
ROLE_ROLE_PRIVS	Роли, назначенные другим ролям.
ROLE_SYS_PRIVS	Системные привилегии, выданные ролям.
ROLE_TAB_PRIVS	Привилегии доступа к таблицам, выданные ролям.
ROLE_COL_PRIVS	Привилегии доступа к столбцам таблиц, выданные ролям.
USER_ROLE_PRIVS	Роли, назначенные текущему пользователю.
USER_SYS_PRIVS	Системные привилегии, выданные текущему пользователю.
USER_TAB_PRIVS	Привилегии доступа к таблицам, выданные текущему пользователю.



# Вопросы?

---



# Администрирование баз данных и приложений

**Архитектура ORACLE: Внешняя память**

Лекция 3

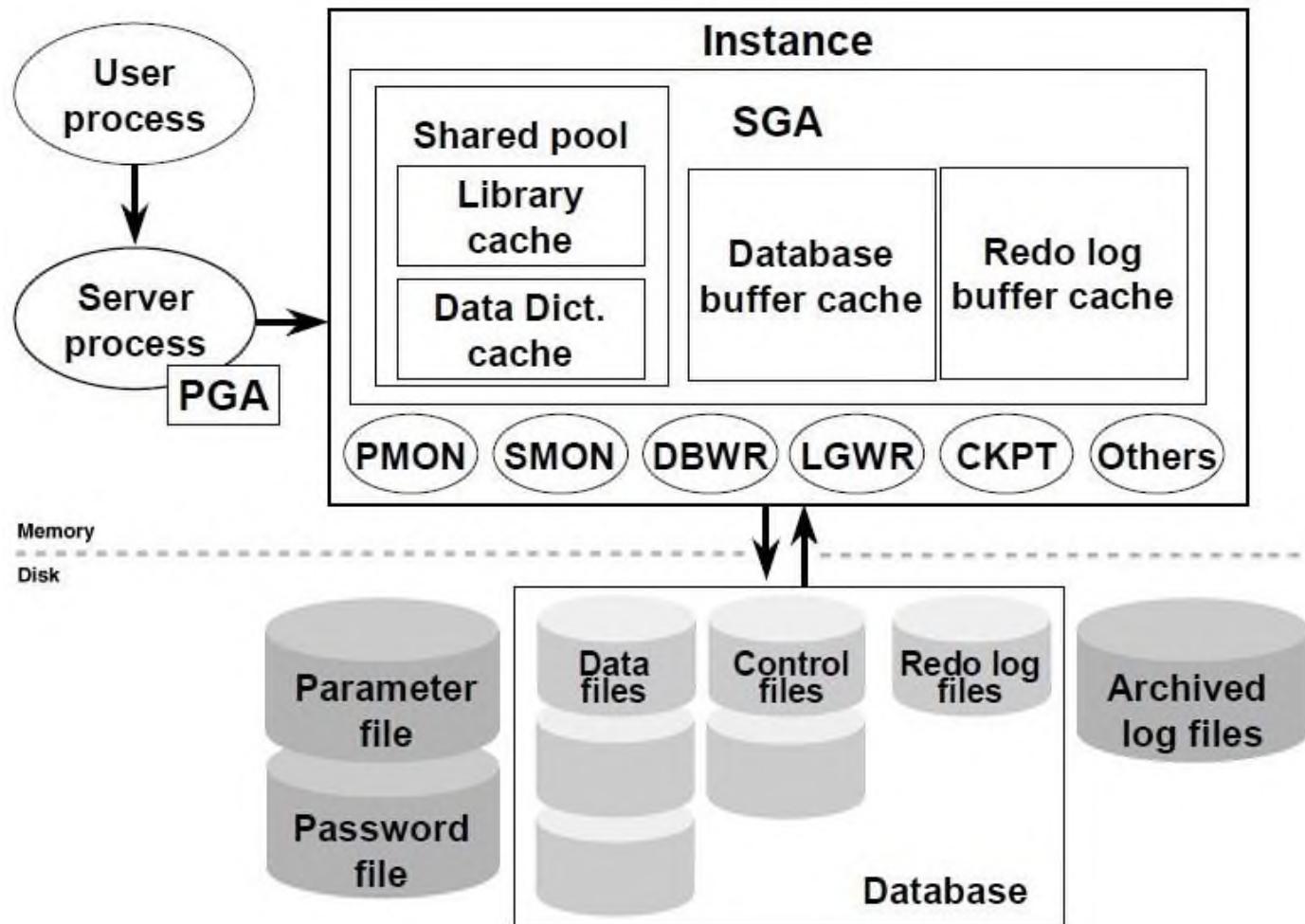
# Архитектура Oracle – основные термины

---

- ▶ **База данных** - набор физических файлов операционной системы
- ▶ **Экземпляр** - набор процессов Oracle и область SGA



# Схема архитектуры Oracle



# Основные компоненты архитектуры Oracle

---

- ▶ **Файлы**, образующие базу данных и поддерживающие экземпляр - файлы параметров, сообщений, данных, временных данных и журналов повторного выполнения
- ▶ **Структуры памяти** - системная глобальная область (System Global Area — SGA) и входящие в SGA пулы
- ▶ **Физические процессы или потоки** - серверные процессы, фоновые процессы и подчиненные процессы



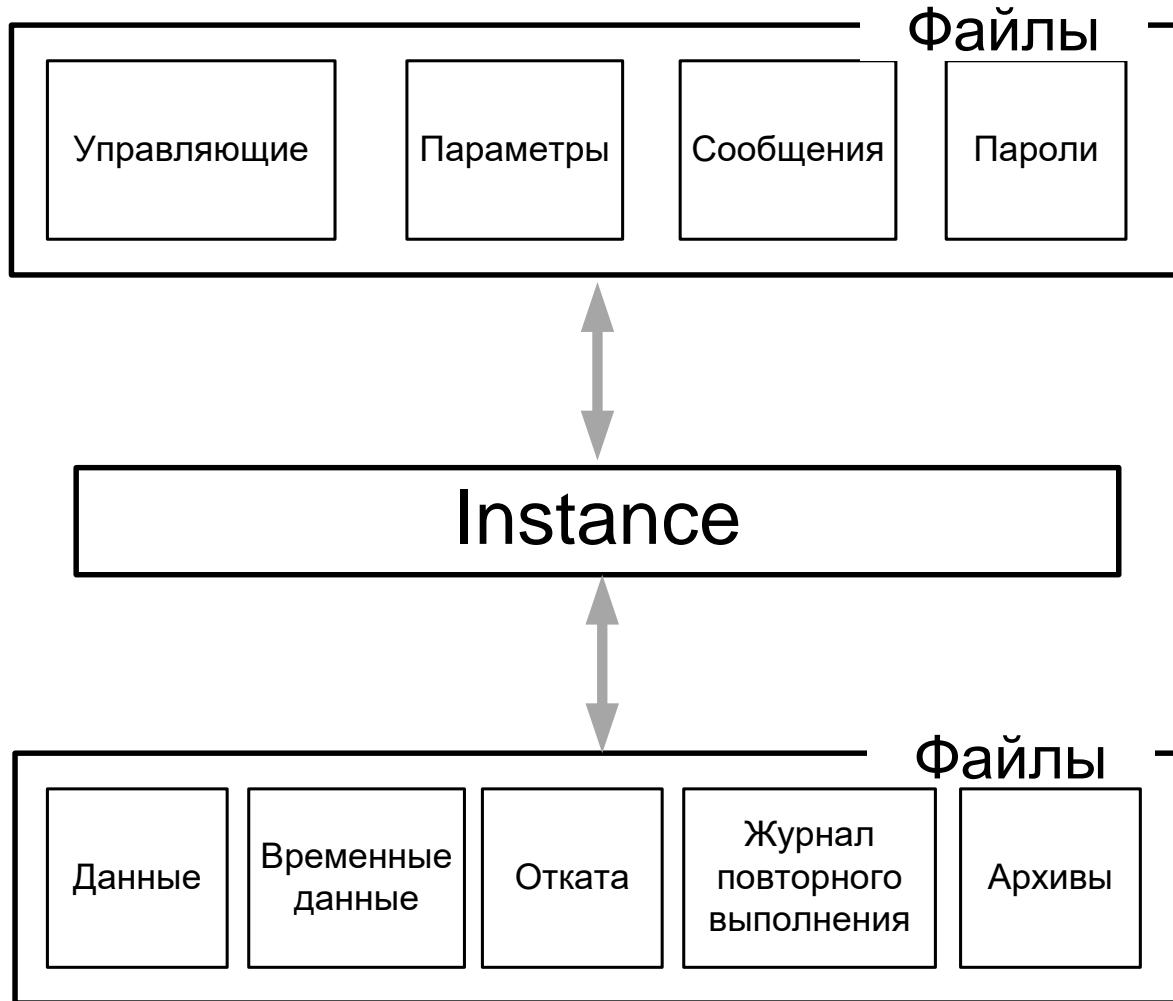
# Архитектура внешней памяти

---

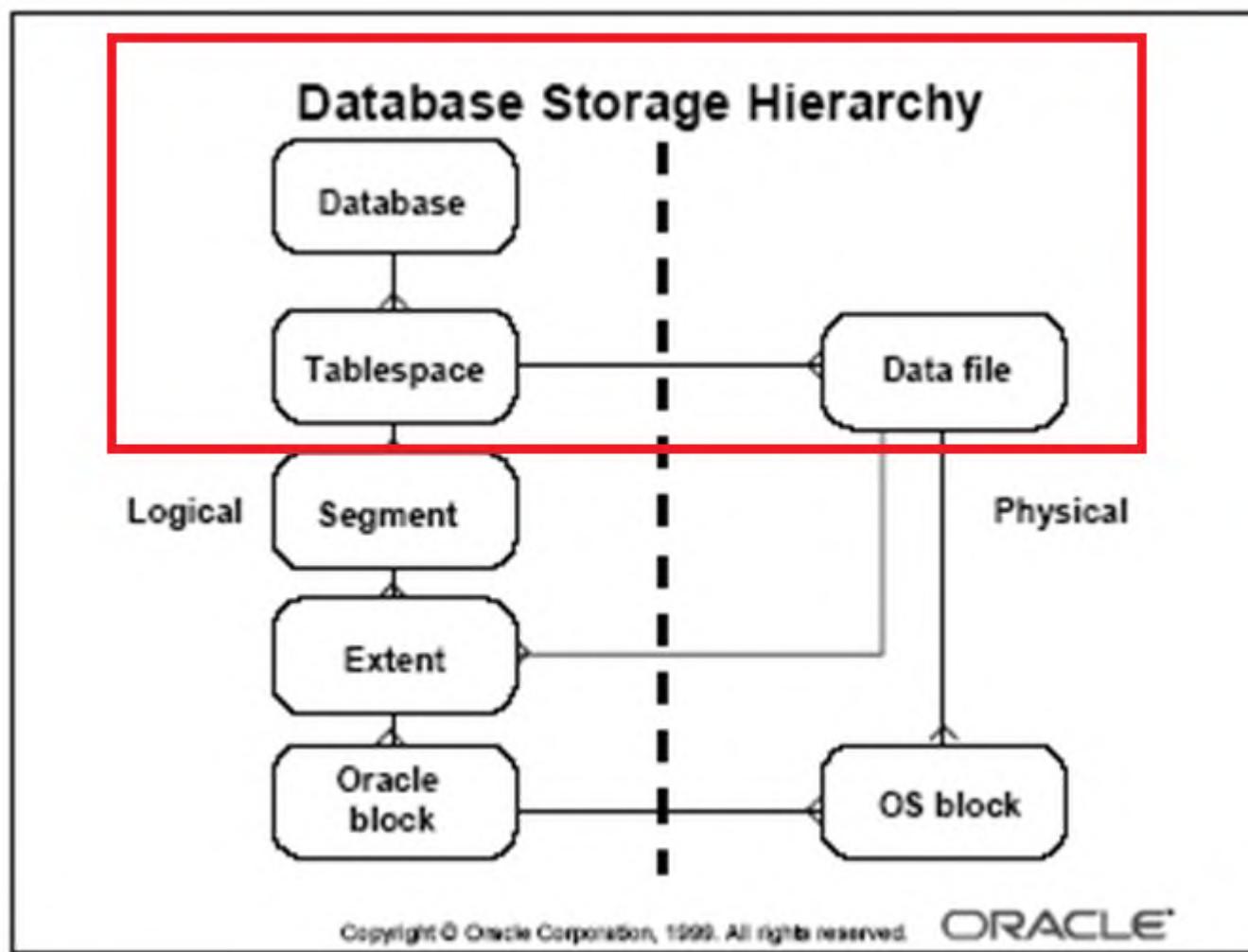
- ▶ **Экземпляр**
  - ▶ Файлы параметров
- ▶ **База данных**
  - ▶ Файлы данных
  - ▶ Файлы журнала повторного выполнения
  - ▶ Управляющие файлы
  - ▶ Временные файлы
  - ▶ Файлы паролей



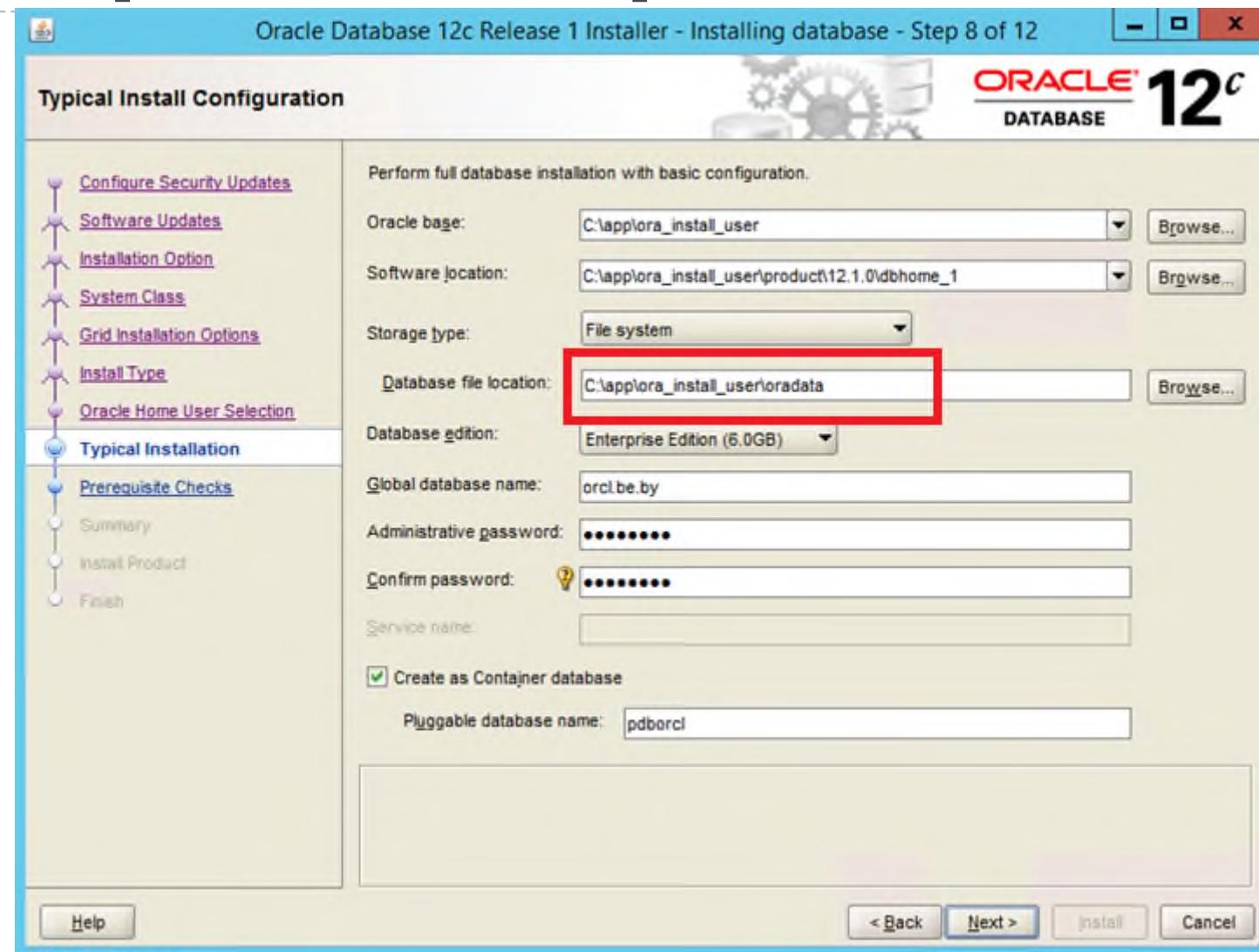
# Архитектура внешней памяти



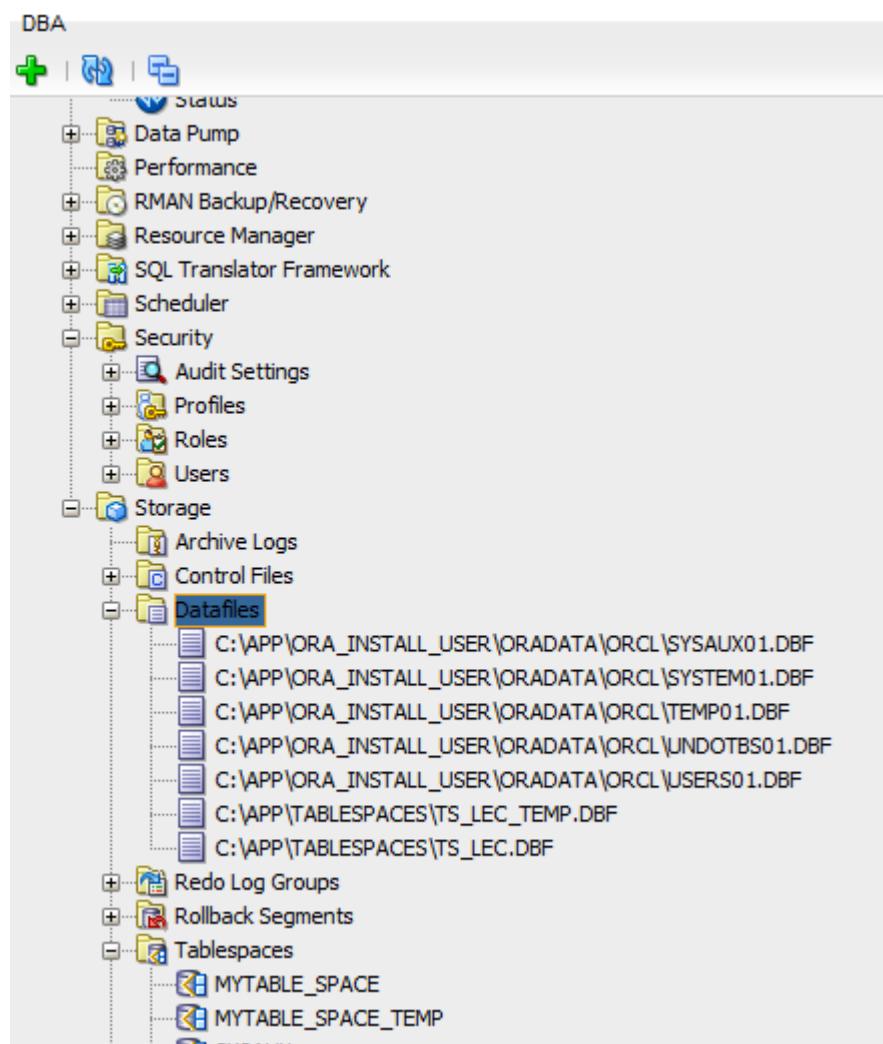
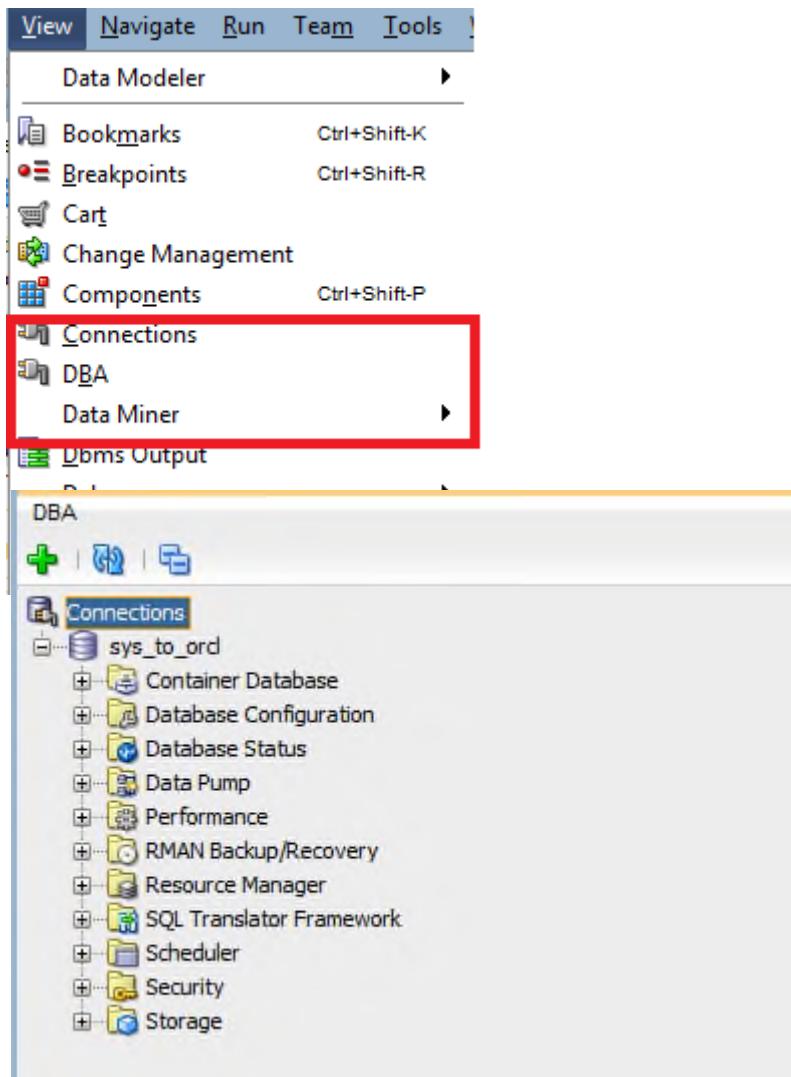
# Схема архитектуры внешней памяти



# Месторасположение файлов базы данных



# Месторасположение файлов базы данных



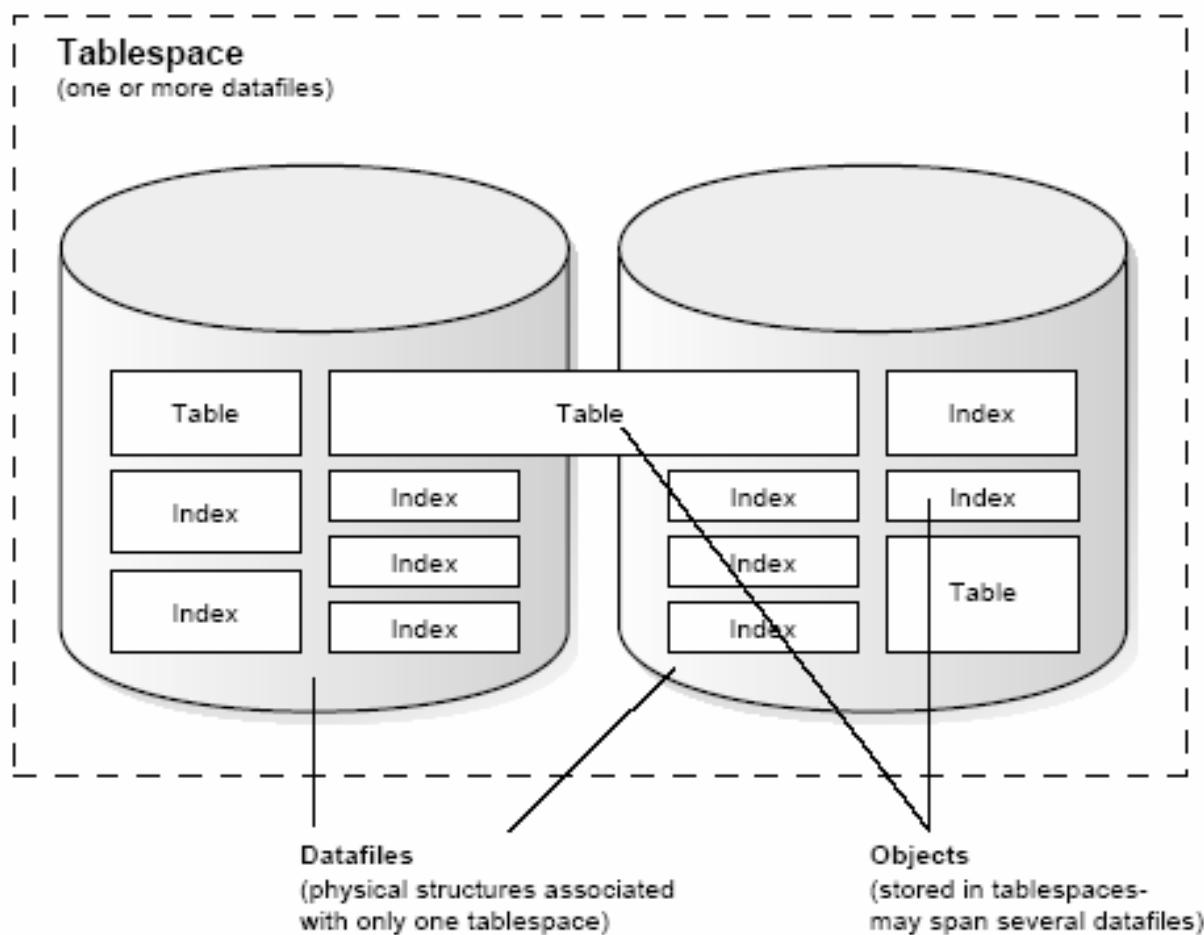
# Табличное пространство

---

- ▶ Табличное пространство – логическая структура хранения данных, контейнер сегментов.
- ▶ Табличное пространство – пограничный объект базы данных: с одной стороны логическая структура сервера, с другой файл или файлы операционной системы. С одним табличным пространством связаны один или несколько файлов, с каждым файлом связано только одно табличное пространство.
- ▶ Данные, временные данные, данные отката – организованы в виде табличных пространств.



# Табличное пространство



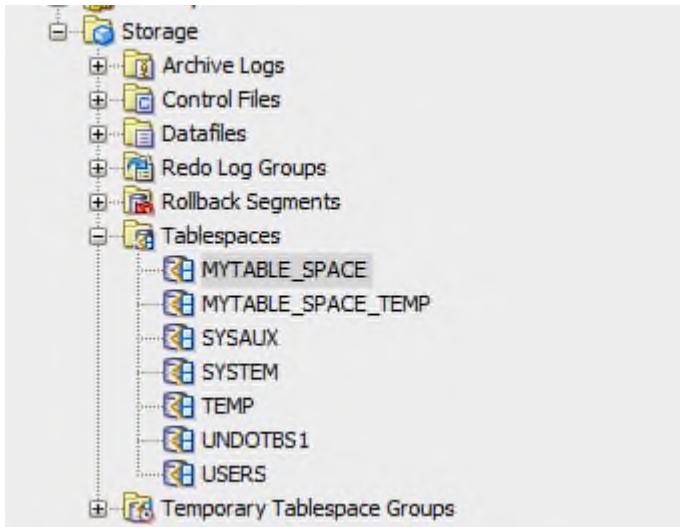
# Предопределенные табличные пространства

---

- ▶ **SYSTEM** используется для управления БД, содержит словарь базы данных, стандартные пакеты процедур.
- ▶ **SYSAUX** – вспомогательное табличное пространство.
- ▶ **TEMP** – временное табличное пространство по умолчанию.
- ▶ **UNDOTBS1** – табличное пространство отката.
- ▶ **USERS** – хранение пользовательских объектов и данных.
- ▶ **EXAMPLE** – демонстрационные схемы.



# Табличные пространства



Name	Value
1 TABLESPACE_NAME	MYTABLE_SPACE
2 BLOCK_SIZE	8192
3 INITIAL_EXTENT	65536
4 NEXT_EXTENT	(null)
5 MIN_EXTENTS	1
6 MAX_EXTENTS	2147483645
7 MAX_SIZE	2147483645
8 PCT_INCREASE	(null)
9 MIN_EXTLEN	65536
10 STATUS	ONLINE
11 CONTENTS	PERMANENT
12 LOGGING	LOGGING
13 FORCE_LOGGING	NO
14 EXTENT_MANAGEMENT	LOCAL
15 ALLOCATION_TYPE	SYSTEM
16 PLUGGED_IN	NO
17 SEGMENT_SPACE_MANAGEMENT	AUTO
18 DEF_TAB_COMPRESSION	DISABLED
19 RETENTION	NOT APPLY
20 BIGFILE	NO
21 PREDICATE_EVALUATION	HOST
22 ENCRYPTED	NO
23 COMPRESS_FOR	(null)

SELECT \* FROM DBA\_TABLESPACES;

# Виды табличных пространств

---

- ▶ **PERMANENT** – предназначены для хранения постоянных объектов (таблиц, индексов, кластеров, ...)
- ▶ **TEMPORARY** – для временных данных
- ▶ **UNDO** – для отката



# Создание табличного пространства

```
CREATE SMALLFILE TABLESPACE TBLSPC_1
  DATAFILE
    'C:\app\Tablespaces\DTBF_1.DBF' SIZE 10M REUSE AUTOEXTEND ON NEXT 2M MAXSIZE 100M ,
    'C:\app\Tablespaces\DTBF_1.DBF' SIZE 10M REUSE AUTOEXTEND ON NEXT 2M MAXSIZE 100M
  BLOCKSIZE 8192
  LOGGING
  ONLINE
  EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;
```

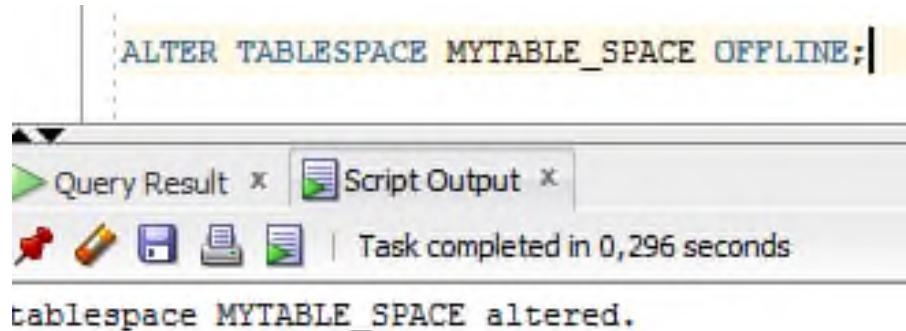
```
SELECT TABLESPACE_NAME, STATUS, CONTENTS, LOGGING FROM DBA_TABLESPACES;
```

Query Result | SQL | All Rows Fetched: 7 in 0,047 seconds

TABLESPACE_NAME	STATUS	CONTENTS	LOGGING
1 SYSTEM	ONLINE	PERMANENT	LOGGING
2 SYSAUX	ONLINE	PERMANENT	LOGGING
3 UNDOTBS1	ONLINE	UNDO	LOGGING
4 TEMP	ONLINE	TEMPORARY	NOLOGGING
5 USERS	ONLINE	PERMANENT	LOGGING
6 MYTABLE_SPACE	ONLINE	PERMANENT	LOGGING
7 MYTABLE_SPACE_TEMP	ONLINE	TEMPORARY	NOLOGGING

# Параметры табличного пространства

- ▶ **SMALLFILE** – 1022 файлов, по  $2^{22}$  блоков
- ▶ **BIGFILE** – 1 файл, 128TB(блок 32K) или 32TB(блок 8K)
- ▶ **LOGGING / NOLOGGING / FORCE LOGGING**
- ▶ **ONLINE / OFFLINE**
- ▶ **REUSE**



The screenshot shows a SQL developer interface. In the top query editor window, the command `ALTER TABLESPACE MYTABLE_SPACE OFFLINE;` is typed. Below it, the results window shows the message `tablespace MYTABLE_SPACE altered.`. The interface includes tabs for 'Query Result' and 'Script Output', and a toolbar with various icons.

```
ALTER TABLESPACE MYTABLE_SPACE OFFLINE;
tablespace MYTABLE_SPACE altered.
```

# Создание таблиц

- ▶ Можно указать табличное пространство.

```
CREATE TABLE XXX1 (X1 INT) TABLESPACE MYTABLE_SPACE;
```

- ▶ На это табличное пространство у пользователя создающего таблицу должна быть достаточная квота.
- ▶ Если tablespace не указан в create table, то таблица будет создана в табличном пространстве по умолчанию
- ▶ Удаление таблиц

```
DROP TABLE XXX1 PURGE;
```



# Табличное пространство для временных данных

---

- ▶ Создается **CREATE TEMPORARY TABLESPACE**
- ▶ Используются для хранения промежуточных результатов сортировки
- ▶ Может быть одно общее для всех PDB (создается при инсталляции в CDB)
- ▶ Может быть создано несколько



# Временное табличное пространство

```
CREATE TEMPORARY TABLESPACE MYTABLE_SPACE_TEMP  
TEMPFILE  
  'C:\APP\TABLESPACES\TS_LEC_TEMP.DBF'  
  SIZE 5M  
  AUTOEXTEND ON NEXT 1M  
  MAXSIZE 26M  
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K;
```

# Табличное пространство для отката UNDO

- ▶ Создается CREATE UNDO TABLESPACE
- ▶ Может быть несколько, но используется только одно
- ▶ Указывается в файле параметров экземпляра

```
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='C:\app\oracle\oradata\ISUS\control01.ctl','C:\app\ISUS\control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='ISUS'
*.db_recovery_file_dest='C:\app\oracle\fast_recovery_area'
*.db_recovery_file_dest_size=6930m
*.diagnostic_dest='C:\app\oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ISUSXDB)'
*.enable_pluggable_database=true
*.memory_target=1638m
*.open_cursors=300
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

# Табличные пространства в представлениях словаря

```
SQL> connect pdb_a_admin/Pa$$w0rd@//localhost:1521/pdb_a.be.by as sysdba;
Connected.
SQL> select tablespace_name, contents from dba tablespaces;

TABLESPACE_NAME          CONTENTS
-----                  -----
SYSTEM                   PERMANENT
SYSAUX                  PERMANENT
TEMP                     TEMPORARY
USERS                   PERMANENT
TBLSP_EA                PERMANENT
TBLSP_TEMP_EA           TEMPORARY
TBLSPC_1                PERMANENT
TBLSPC_2                PERMANENT

8 rows selected.

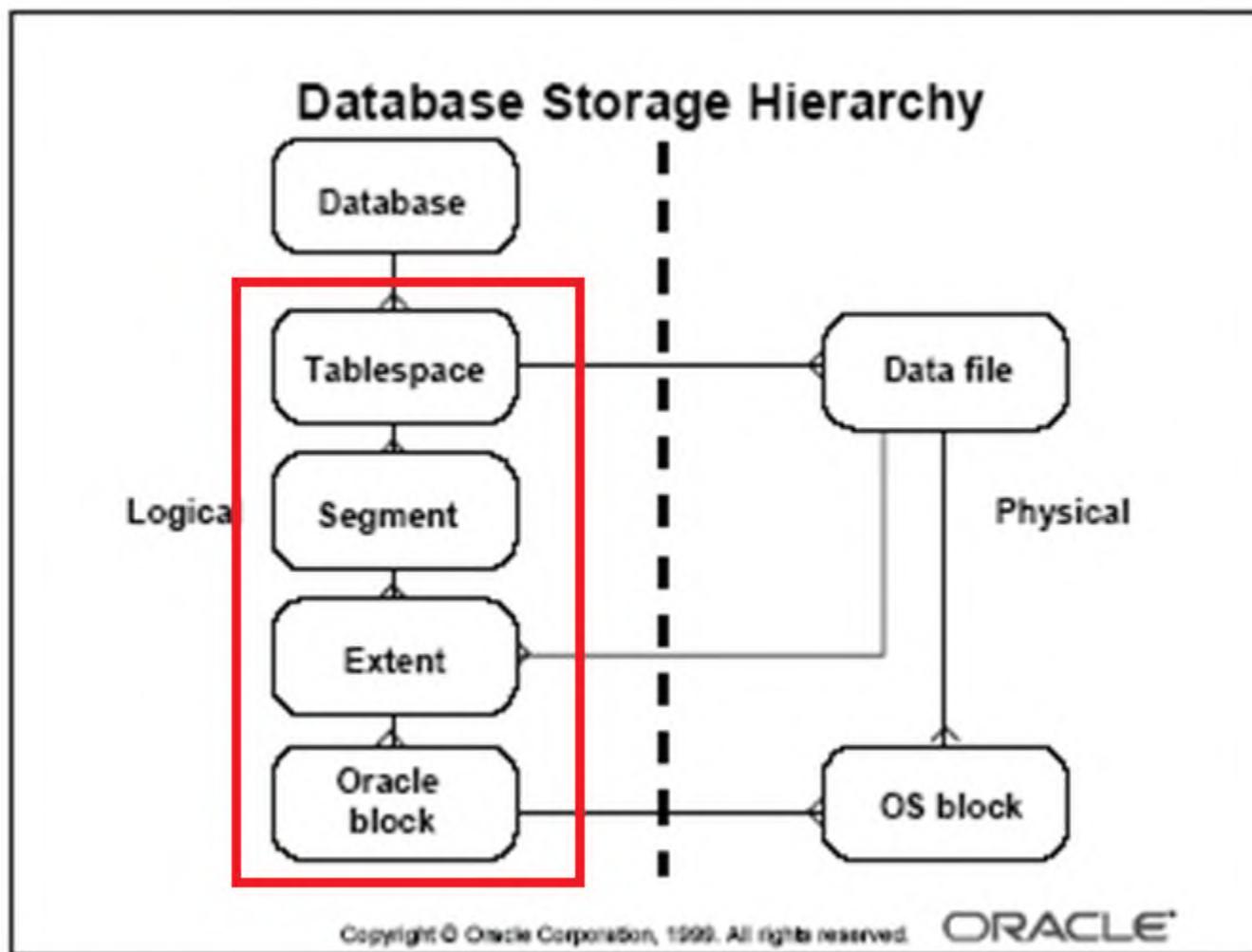
SQL> connect /as sysdba;
Connected.
SQL> select tablespace_name, contents from dba tablespaces;

TABLESPACE_NAME          CONTENTS
-----                  -----
SYSTEM                   PERMANENT
SYSAUX                  PERMANENT
UNDOTBS1                UNDO
TEMP                     TEMPORARY
USERS                   PERMANENT
MYTABLE_SPACE            PERMANENT
MYTABLE_SPACE_TEMP       TEMPORARY

7 rows selected.
```



# Схема архитектуры внешней памяти



# Сегмент

- ▶ Сегмент – область на диске, выделяемая под объекты.
- ▶ Сегменты типизируются в зависимости от типа данных, хранящихся в них – сегменты таблиц, сегменты индексов, сегменты кластеров и т.д.(всего 10 типов).

```
select distinct segment_type from dba_segments;
```

SEGMENT_TYPE
1 LOBINDEX
2 INDEX PARTITION
3 ROLLBACK
4 NESTED TABLE
5 TABLE PARTITION
6 LOB PARTITION
7 LOBSEGMENT
8 INDEX
9 TABLE
10 CLUSTER

# Экстент

---

- ▶ Экстент – непрерывный фрагмент дисковой памяти.
- ▶ Является единицей выделения вторичной памяти (выделяется целым числом экстентов).
- ▶ Когда экстент заполняется выделяется следующий.
- ▶ Размер экстента варьируется от одного блока до 2 Гб.



# БЛОКИ

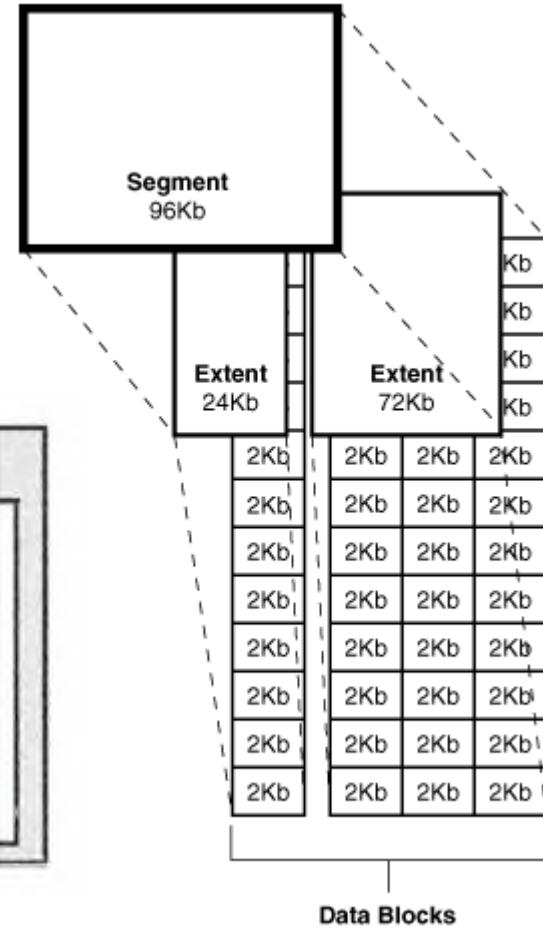
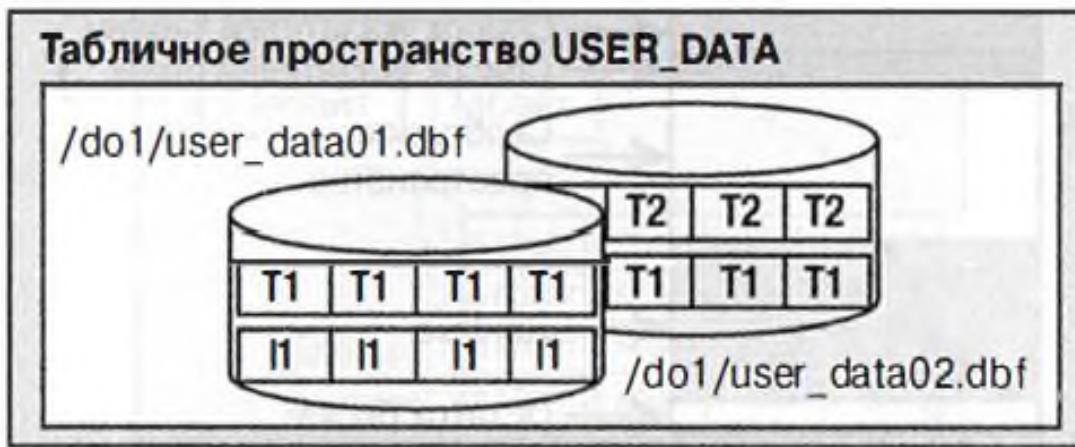
- ▶ Блок – минимальная единица объема памяти, применяемая при записи и чтении данных.
  - ▶ Размер кратен 2К, и должен быть кратен величине блока операционной системы (2К, 4К, 8К, допустимы 16К, 32К).
  - ▶ Устанавливается в файле параметров экземпляра при создании БД,
  - ▶ В табличном пространстве все блоки одного размера.



## Схема блока

# Сегмент – экстент – блок

- ▶ Сегмент состоит из одного и более экстентов
- ▶ Экстент состоит из идущих подряд блоков



# Сегмент

---

- ▶ Располагается в табличном пространстве.
- ▶ В одном табличном пространстве может быть много сегментов.
- ▶ Сегмент, если он не секционирован, располагается в одном табличном пространстве.
- ▶ Управление размерностью сегментов может осуществляться автоматически (*segment space management auto*) или вручную (*segment space management manual*).



# Сегмент

```
select tablespace_name,          -- имя табличного пространства
       block_size,            -- размер блок
       initial_extent,        -- размер экстента
       initial_extent/block_size, -- кол. блоков в экстенте
       extent_management,     -- управление экстентами
       segment_space_management, -- управление сегментами
       bigfile
  from dba_tablespaces;
```

#	TABLESPACE_NAME	BLOCK_SIZE	INITIAL_EXTENT	INITIAL_EXTENT/BLOCK_SIZE	EXTENT_MANAGEMENT	SEGMENT_SPACE_MANAGEMENT	BIGFILE
1	SYSTEM	8192	65536		8 LOCAL	MANUAL	NO
2	SYSAUX	8192	65536		8 LOCAL	AUTO	NO
3	TEMP	8192	1048576		128 LOCAL	MANUAL	NO
4	USERS	8192	65536		8 LOCAL	AUTO	NO
5	SVV_TBS01	8192	65536		8 LOCAL	AUTO	NO
6	SVV_TBS02	8192	65536		8 LOCAL	AUTO	NO

# Сегмент

- ▶ сразу после создания табличного пространства в нем нет сегментов

```
select case
    when exists (select* from dba_segments where tablespace_name='SVV_TBS01') then 'да'
    else 'нет'
    end as "сегменты есть в SVV_TBS01?"
from dual;
```

	сегменты есть в SVV_TBS01?
1	нет

# Сегмент

- ▶ Сразу после создания таблицы сегмент не создается
- ▶ Deferred segment create – отложенное создание сегментов

```
create table XXX01(x int) tablespace SVV_TBS01;
select table_name, tablespace_name from user_tables;


| TABLE_NAME | TABLESPACE_NAME |
|------------|-----------------|
| 1 XXX01    | SVV_TBS01       |


select case
    when exists (select* from dba_segments where tablespace_name='SVV_TBS01') then 'да'
    else 'нет'
    end as "сегменты есть в SVV_TBS01?"
from dual;

select case
    when exists (select* from dba_segments where tablespace_name='SVV_TBS01') then 'да'
    else 'нет'
    end as "сегменты есть в SVV_TBS01?"
from dual;
```

сегменты есть в SVV_TBS01?
1 нет

```
show parameter segment
-----  
deferred_segment_creation          boolean      TRUE  
rollback_segments                  string  
transactions_per_rollback_segment integer      5
```

# Сегмент

- ▶ Сегмент хранит только данные, поэтому он создается только при добавлении данных (в примере строки в таблицу).

```
insert into XXX01 values(1);
commit;

select owner, segment_name, segment_type, tablespace_name, bytes, blocks, buffer_pool  from dba_segments
where tablespace_name='SVV_TBS01';
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	BUFFER_POOL
ISUS_B	XXX01	TABLE	SVV_TBS01	65536	8	DEFAULT

# Сегмент

- ▶ При удалении строк (delete) из таблицы, сегмент не удаляется.

```
delete XXX01;  
commit;
```

```
select owner, segment_name, segment_type, tablespace_name, bytes, blocks, buffer_pool  from dba_segments  
where tablespace_name='SVV_TBS01';
```

	OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	BUFFER_POOL
1	ISUS_B	XXX01	TABLE	SVV_TBS01	65536	8	DEFAULT

# Сегмент

- При удалении таблицы (drop table) изменяется имя сегмента, и информация об удалении записывается в словарь базы данных.

```
drop table XXX01;
select owner, segment_name, segment_type, tablespace_name, bytes, blocks, buffer_pool from dba_segments
where tablespace_name='SVV_TBS01';
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	BUFFER_POOL
1 ISUS_B	BIN\$BhPbprm+TV2KJUDB20zYvQ==\$0	TABLE	SVV_TBS01	65536	8	DEFAULT

```
select object_name, original_name, operation, type, ts_name, createtime, droptime from user_recyclebin;
```

OBJECT_NAME	ORIGINAL_NAME	OPERATION	TYPE	TS_NAME	CREATETIME	DROPTIME
1 BIN\$BhPbprm+TV2KJUDB20zYvQ==\$0	XXX01	DROP	TABLE	SVV_TBS01	2015-09-25:14:24:05	2015-09-25:15:09:50

# Сегмент

- ▶ Таблица и ее содержимое (сегмент) могут быть восстановлены с помощью механизма RECYCLEBIN.

```
flashback table XXX01 to before drop;  
select owner, segment_name, segment_type, tablespace_name, bytes, blocks, buffer_pool from dba_segments  
where tablespace_name='SVV_TBS01';
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	BUFFER_POOL
ISUS_B	XXX01	TABLE	SVV_TBS01	65536	8	DEFAULT

```
select table_name, tablespace_name from user_tables;
```

TABLE_NAME	TABLESPACE_NAME
XXX01	SVV_TBS01

# Сегмент

- ▶ Для удаления RECYCLEBIN-сегмента применяется команда PURGE.

```
select * from user_recyclebin;
```

OBJECT_NAME	ORIGINAL_NAME	OPERATION	TYPE	TS_NAME	CREATETIME	DROPTIME
1 BIN\$j19X3qN3Q0WTQ+5K8nRaow==\$0	XXX02	DROP	TABLE	SVV_TBS01	2015-09-25:19:35:08	2015-09-25:19:35:16

```
select segment_name, segment_type, tablespace_name, bytes, blocks, buffer_pool from user_segments
```

SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	BUFFER_POOL
1 BIN\$j19X3qN3Q0WTQ+5K8nRaow==\$0	TABLE	SVV_TBS01	65536	8	KEEP

```
purge table XXX02;
```

# Сегмент

- Если при удалении объекта применяется опция PURGE, то RECYCLEBIN-сегмент не сохраняется и восстановление объекта невозможно.

```
drop table XXX01 purge;

select case
    when exists (select* from dba_segments where tablespace_name='SVV_TBS01') then 'да'
    else 'нет'
    end as "сегменты есть в SVV_TBS01?"
from dual;
```

сегменты есть в SVV_TBS01?	
1	нет

## Экстент

---

- ▶ Длину выделяемого экстента вычисляет СУБД (не меньше 64К).
- ▶ Если при создании табличного пространства задана опция **UNIFORM**, то все экстенты имеют одинаковую длину.
- ▶ Управление экстентами возможно локальное **LOCAL** и через словарь базы данных **DICTIONARY**.



# Экстент - пример

```
create smallfile                                -- SMALLFILE (ум) / BIGFILE
    tablespace SVV_TBS03
    datafile 'c:\app\oracle\oradata\SVV_TBS03.DBF'      -- перманентное пространство
    size 10m reuse autoextend on next 5m maxsize 1024m -- размеры
    uniform                                         -- все экстенты одного размера
    logging                                         -- журналирование/nologging (ум)
    online                                          -- offline/online(ум)
    extent management local                      -- локальное управление экстентами
    segment space management auto;                -- ручное управление сегментами

create table XXX03(x int)  tablespace  SVV_TBS03; ;

begin
    delete XXX03;
    for k in 1..1000000
    loop
        insert into XXX03(x) values(k);
    end loop;
    commit;
end;

select segment_name, segment_type, tablespace_name, bytes, blocks, extents, buffer_pool
from user_segments
where tablespace_name='SVV_TBS03';
```

	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	EXTENTS	BUFFER_POOL
1	XXX03	TABLE	SVV_TBS03	13631488	1664	13	DEFAULT

# Экстент - пример

```
select * from user_extents where tablespace_name = 'SVV_TBS03';
```

	SEGMENT_NAME	PARTITION_NAME	SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	BYTES	BLOCKS
1	XXX03	(null)	TABLE	SVV_TBS03	0	1048576	128
2	XXX03	(null)	TABLE	SVV_TBS03	1	1048576	128
3	XXX03	(null)	TABLE	SVV_TBS03	2	1048576	128
4	XXX03	(null)	TABLE	SVV_TBS03	3	1048576	128
5	XXX03	(null)	TABLE	SVV_TBS03	4	1048576	128
6	XXX03	(null)	TABLE	SVV_TBS03	5	1048576	128
7	XXX03	(null)	TABLE	SVV_TBS03	6	1048576	128
8	XXX03	(null)	TABLE	SVV_TBS03	7	1048576	128
9	XXX03	(null)	TABLE	SVV_TBS03	8	1048576	128
10	XXX03	(null)	TABLE	SVV_TBS03	9	1048576	128
11	XXX03	(null)	TABLE	SVV_TBS03	10	1048576	128
12	XXX03	(null)	TABLE	SVV_TBS03	11	1048576	128
13	XXX03	(null)	TABLE	SVV_TBS03	12	1048576	128

# Экстент - пример удаления таблицы

```
delete xxx03;  
commit;  
select * from user_extents where tablespace_name = 'SVV_TBS03';
```

	SEGMENT_NAME	PARTITION_NAME	SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	BYTES	BLOCKS
1	XXX03	(null)	TABLE	SVV_TBS03	0	1048576	128
2	XXX03	(null)	TABLE	SVV_TBS03	1	1048576	128
3	XXX03	(null)	TABLE	SVV_TBS03	2	1048576	128
4	XXX03	(null)	TABLE	SVV_TBS03	3	1048576	128
5	XXX03	(null)	TABLE	SVV_TBS03	4	1048576	128
6	XXX03	(null)	TABLE	SVV_TBS03	5	1048576	128
7	XXX03	(null)	TABLE	SVV_TBS03	6	1048576	128
8	XXX03	(null)	TABLE	SVV_TBS03	7	1048576	128
9	XXX03	(null)	TABLE	SVV_TBS03	8	1048576	128
10	XXX03	(null)	TABLE	SVV_TBS03	9	1048576	128
11	XXX03	(null)	TABLE	SVV_TBS03	10	1048576	128
12	XXX03	(null)	TABLE	SVV_TBS03	11	1048576	128
13	XXX03	(null)	TABLE	SVV_TBS03	12	1048576	128

# Экстент - сжатие

```
alter table XXX03 enable row movement;      -- разрешить перемещение строк таблицы  
alter table XXX03 shrink space cascade;    -- удалить неиспользуемые экстенты
```

```
select * from user_extents where tablespace_name = 'SVV_TBS03';
```

	SEGMENT_NAME	PARTITION_NAME	SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	BYTES	BLOCKS
1	XXX03	(null)	TABLE	SVV_TBS03	0	1048576	128

```
select segment_name, segment_type, tablespace_name, bytes, blocks, extents, buffer_pool  
from user_segments where tablespace_name = 'SVV_TBS03';
```

	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BYTES	BLOCKS	EXTENTS	BUFFER_POOL
1	XXX03	TABLE	SVV_TBS03	1048576	128	1	DEFAULT

# БЛОК

---

- ▶ Устанавливается в файле параметров экземпляра.

```
show parameter block;
```

NAME	TYPE	VALUE
db_block_buffers	integer	0
db_block_checking	string	FALSE
db_block_checksum	string	TYPICAL
db_block_size	integer	8192
db_file_multiblock_read_count	integer	128



# БЛОК

```
select file_name,          -- имя файла
       tablespace_name,    -- имя табличного пространства
       maxblocks,           -- максимальное количество блоков
       user_blocks         -- используется блоков
      from dba_data_files;
```

	FILE_NAME	TABLESPACE_NAME	MAXBLOCKS	USER_BLOCKS
1	C:\APP\ORACLE\ORADATA\ISUS\ISUS_B\SYSTEM01.DBF	SYSTEM	4194302	33152
2	C:\APP\ORACLE\ORADATA\ISUS\ISUS_B\SYSAux01.DBF	SYSAUX	4194302	80512
3	C:\APP\ORACLE\ORADATA\ISUS\ISUS_B\ISUS_B_USERS01.DBF	USERS	4194302	512
4	C:\APP\ORACLE\ORADATA\SVV_TBS01.DBF	SVV_TBS01	131072	1792
5	C:\ISUS_B_TBS\SVV_TBS02.DBF	SVV_TBS02	131072	1152
6	C:\APP\ORACLE\ORADATA\SVV_TBS03.DBF	SVV_TBS03	131072	1792

```
select file_name,          -- имя файла
       tablespace_name,    -- имя табличного пространства
       maxblocks,           -- максимальное количество блоков
       user_blocks         -- используется блоков
      from dba_temp_files ;
```

	FILE_NAME	TABLESPACE_NAME	MAXBLOCKS	USER_BLOCKS
1	C:\APP\ORACLE\ORADATA\ISUS\ISUS_B\PDBSEED_TEMP01.DBF	TEMP	4194302	60416

# Блок в файле параметров

```
isus.__data_transfer_cache_size=0
isus.__db_cache_size=570425344
isus.__java_pool_size=16777216
isus.__large_pool_size=33554432
isus.__oracle_base='C:\app\oracle'#ORACLE_BASE set from environment
isus.__pga_aggregate_target=603979776
isus.__sga_target=1124073472
isus.__shared_io_pool_size=50331648
isus.__shared_pool_size=436207616
isus.__streams_pool_size=0
*.audit_file_dest='C:\app\oracle\admin\ISUS\adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='C:\app\oracle\oradata\ISUS\control01.ctl','C:\app\oracle\oradata\ISUS\control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='ISUS'
*.db_recovery_file_dest='C:\app\oracle\fast_recovery_area'
*.db_recovery_file_dest_size=6930m
*.diagnostic_dest='C:\app\oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ISUSXDB)'
*.enable_pluggable_database=true
*.memory_target=1638m
*.open_cursors=300
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

# Итого

---

- ▶ База данных состоит из одного или нескольких табличных пространств.
- ▶ Табличное пространство состоит из одного или нескольких файлов данных. Табличное пространство содержит сегменты.
- ▶ Сегменты состоят из экстентов. Сегмент привязан к табличному пространству, но его данные могут находиться в разных файлах данных, образующих это табличное пространство.
- ▶ Экстент — набор расположенных рядом на диске блоков. Экстент целиком находится в одном табличном пространстве и, более того, в одном файле данных этого табличного пространства.
- ▶ Блок — наименьшая единица управления пространством в базе данных. Блок — наименьшая единица ввода-вывода, используемая сервером.



# Вопросы?

---

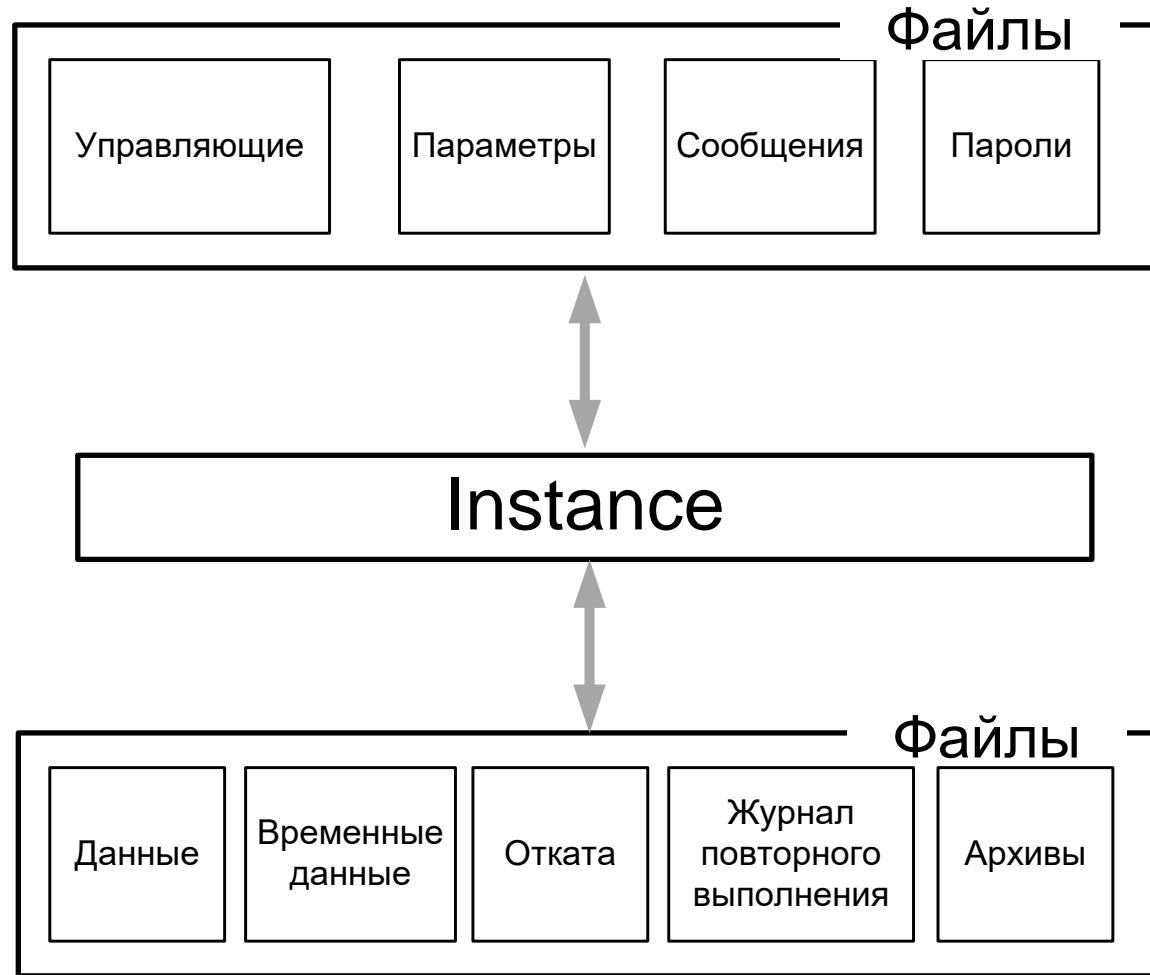


# **Администрирование баз данных и приложений**

**Архитектура ORACLE: Файлы БД и сервера**

**Лекция 4**

# Архитектура внешней памяти



# Файл параметров

- ▶ Файл параметров предназначен для хранения параметров экземпляра

```
connect system
SELECT name, DESCRIPTION FROM V$PARAMETER;
```

Script Output X Query Result X

SQL | Fetched 50 rows in 0,031 seconds

NAME	DESCRIPTION
1 lock_name_space	lock name space used for generating lock names for standby/clone database
2 processes	user processes
3 sessions	user and system sessions
4 timed_statistics	maintain internal timing statistics
5 timed_os_statistics	internal os statistic gathering interval in seconds
6 resource_limit	master switch for resource limit
7 license_max_sessions	maximum number of non-system user sessions allowed
8 license_sessions_warning	warning level for number of non-system user sessions
9 cpu_count	number of CPUs for this instance
10 instance_groups	list of instance group names
11 event	debug event control - default null string
12 sga_max_size	max total SGA size
13 use_large_pages	Use large pages if available (TRUE/FALSE/ONLY)
14 pre_page_sga	pre-page sga for process
15 shared_memory_address	SGA starting address (low order 32-bits on 64-bit platforms)
16 hi_shared_memory_address	SGA starting address (high order 32-bits on 64-bit platforms)

# Файл параметров

## Наименование параметра – значение параметра

connect system

SELECT name, value, description FROM v\$parameter;		
NAME	VALUE	DESCRIPTION
1 lock_name_space	(null)	lock name space used for generating lock names for standby/clone
2 processes	300	user processes
3 sessions	472	user and system sessions
4 timed_statistics	TRUE	maintain internal timing statistics
5 timed_os_statistics	0	internal os statistic gathering interval in seconds
6 resource_limit	FALSE	master switch for resource limit
7 license_max_sessions	0	maximum number of non-system user sessions allowed
8 license_sessions_warning	0	warning level for number of non-system user sessions
9 cpu_count	1	number of CPUs for this instance
10 instance_groups	(null)	list of instance group names
11 event	(null)	debug event control - default null string
12 sga_max_size	1677721600	max total SGA size
13 use_large_pages	TRUE	Use large pages if available (TRUE/FALSE/ONLY)
14 pre_page_sga	TRUE	pre-page sga for process
15 shared_memory_address	0	SGA starting address (low order 32-bits on 64-bit platforms)
16 hi_shared_memory_address	0	SGA starting address (high order 32-bits on 64-bit platforms)
17 use indirect data buffers	FALSE	Enable indirect data buffers (very large SGA on 32-bit platform)

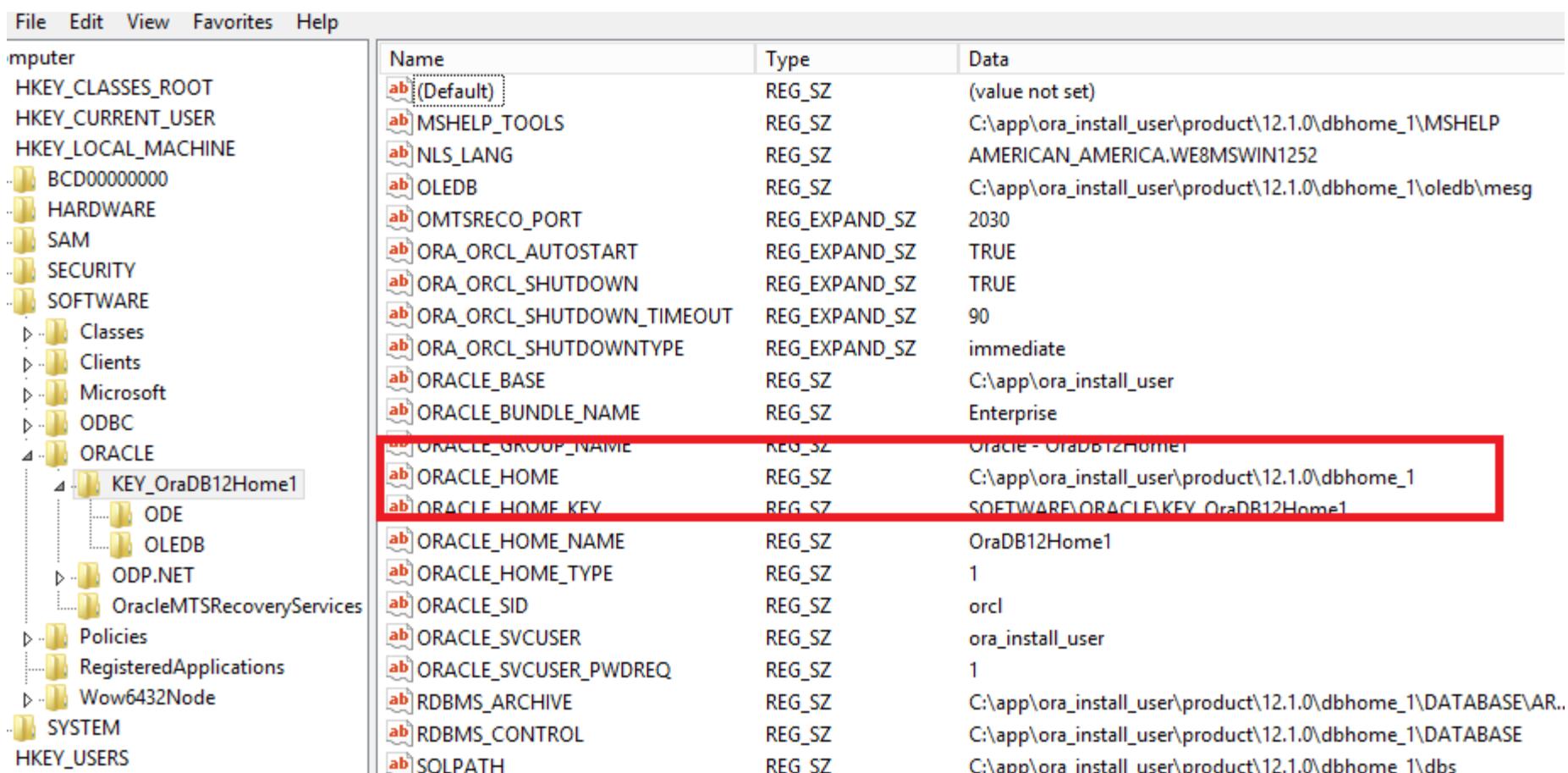
# Файл параметров

- ▶ Месторасположение файла параметров по умолчанию

Platform	Default Location
UNIX and Linux	<i>ORACLE_HOME/dbs</i>
Windows	<i>ORACLE_HOME\database</i>



# Файл параметров

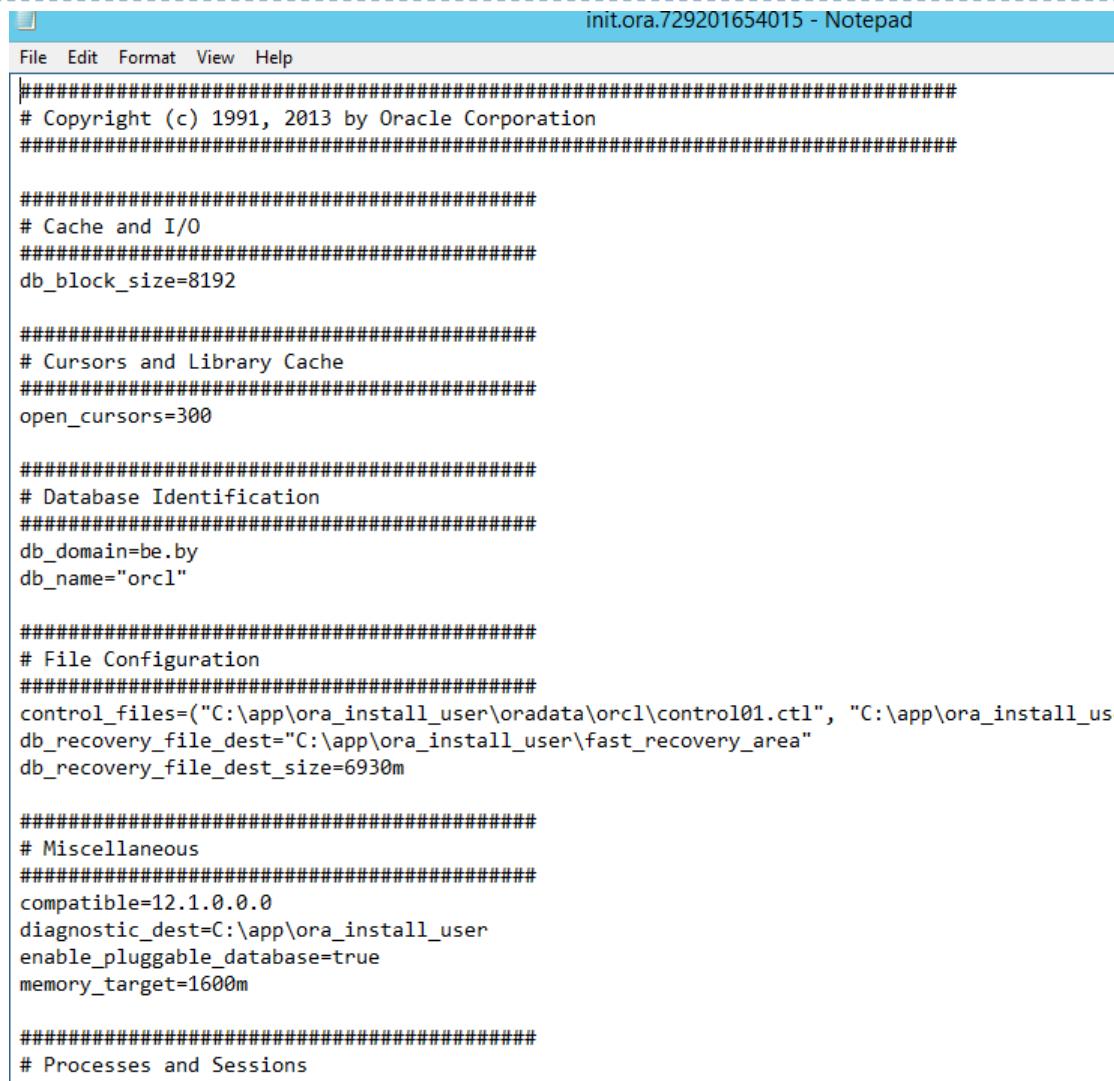


# Файл параметров

Name		Date modified
archive		29.08.2016 5:27
hc_orcl.dat		29.08.2016 5:30
oradba		22.12.2005 4:07
oradim		26.09.2016 4:47
PWDorcl		29.08.2016 5:57
SPFILEORCL		26.09.2016 22:00

Name	Date modified	Type	Size	Folder path
init.ora.729201654015	29.08.2016 5:30	729201654015 ...	2 KB	C:\app\ora_install_user\admin\orcl\pfile
initdg4odbc	28.06.2013 1:09	ORA File	1 KB	C:\app\ora_install_user\product\12.1.0\dbhome_1\hs\admin
init	03.02.2012 10:24	ORA File	3 KB	C:\app\ora_install_user\product\12.1.0\dbhome_1\ dbs
init	25.07.2001 20:04	ORA File	8 KB	C:\app\ora_install_user\product\12.1.0\dbhome_1\srvm\admin

# Файл параметров



The screenshot shows a Windows Notepad window titled "init.ora.729201654015 - Notepad". The window contains the Oracle database initialization parameter file (init.ora). The file is structured with various sections separated by triple-line comments (#####). Key parameters include db\_block\_size=8192, open\_cursors=300, db\_domain=be.by, db\_name="orcl", control\_files, db\_recovery\_file\_dest, db\_recovery\_file\_dest\_size, compatible=12.1.0.0.0, diagnostic\_dest, enable\_pluggable\_database=true, memory\_target=1600m, and processes related parameters.

```
init.ora.729201654015 - Notepad
File Edit Format View Help
#####
# Copyright (c) 1991, 2013 by Oracle Corporation
#####

#####
# Cache and I/O
#####
db_block_size=8192

#####
# Cursors and Library Cache
#####
open_cursors=300

#####
# Database Identification
#####
db_domain=be.by
db_name="orcl"

#####
# File Configuration
#####
control_files=("C:\app\ora_install_user\oradata\orcl\control01.ctl", "C:\app\ora_install_use
db_recovery_file_dest="C:\app\ora_install_user\fast_recovery_area"
db_recovery_file_dest_size=6930m

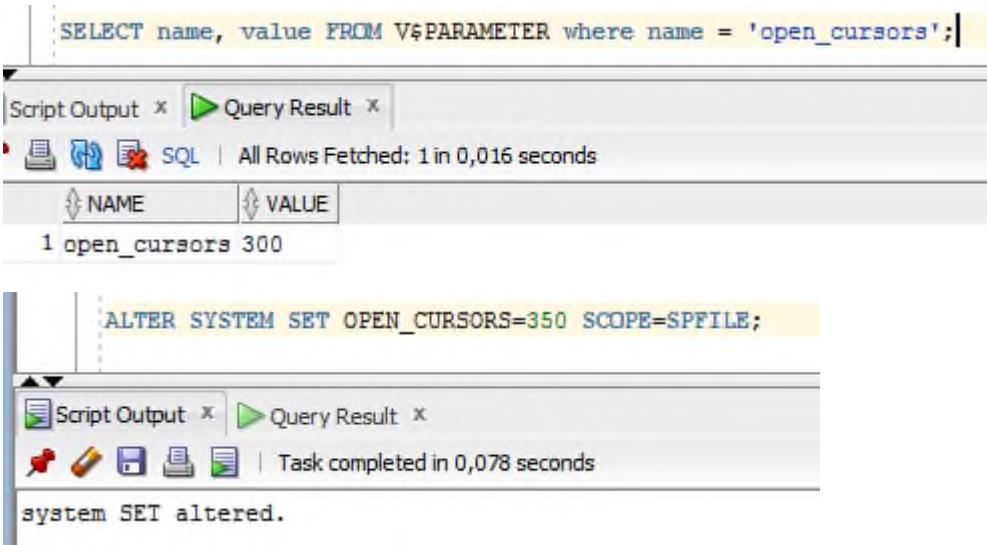
#####
# Miscellaneous
#####
compatible=12.1.0.0.0
diagnostic_dest=C:\app\ora_install_user
enable_pluggable_database=true
memory_target=1600m

#####
# Processes and Sessions
#####
```



# Файл параметров

- ▶ SPFILE - файл параметров сервера в двоичном виде
- ▶ Может изменяться командой ALTER SYSTEM ...  
**SCOPE=SPFILE**
- ▶ Изменение параметров сохраняется в SPFILE и будет применяться при следующем старте Oracle



The screenshot shows two separate sessions in Oracle SQL Developer.

The top session shows the result of the query:

```
SELECT name, value FROM V$PARAMETER where name = 'open_cursors';
```

The result is:

NAME	VALUE
1 open_cursors	300

The bottom session shows the execution of the command:

```
ALTER SYSTEM SET OPEN_CURATORS=350 SCOPE=SPFILE;
```

The result is:

```
system SET altered.
```

# Файл параметров

```
SELECT name, value FROM V$PARAMETER where name = 'open_cursors';

ALTER SYSTEM SET OPEN_CURSORS=350 SCOPE=SPFILE;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0,015 seconds

NAME	VALUE
1 open_cursors	300

```
SQL> connect /as sysdba;
Connected.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup open;
ORACLE instance started.

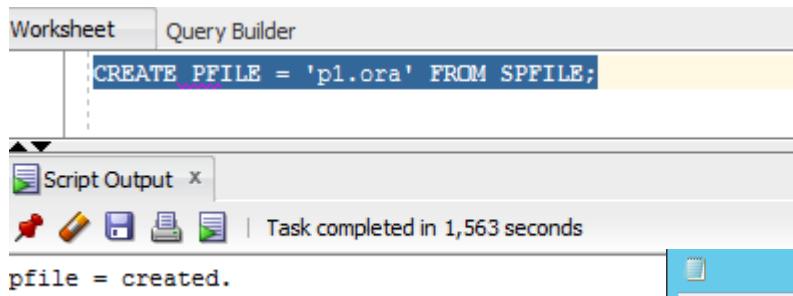
Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers    654311424 bytes
Redo Buffers        6873088 bytes
Database mounted.
Database opened.
SQL> -
```

```
SELECT name, value FROM V$PARAMETER where name = 'open_cursors';
```

Query Result x  
SQL | All Rows Fetched: 1 in 0,031 seconds

NAME	VALUE
1 open_cursors	350

# Файл параметров



Создание текстового файла  
параметров PFILE из бинарного  
файла SPFILE.

The screenshot shows a Windows Notepad window titled 'p1 - Notepad'. The window displays a large block of Oracle database initialization parameters (spfile contents) in plain text. The parameters include settings for data transfer cache size, database cache size, Java pool size, large pool size, oracle base, PGA aggregate target, SGA target, shared pool sizes, streams pool size, audit file destination, audit trail, compatibility, control files, db block size, db domain, db name, db recovery file destination, db recovery file size, diagnostic dest, dispatchers, enable pluggable database, memory target, open cursors, processes, remote login password file, and undo tablespace.

```
File Edit Format View Help
p1 - Notepad
orcl._data_transfer_cache_size=0
orcl._db_cache_size=771751936
orcl._java_pool_size=16777216
orcl._large_pool_size=33554432
orcl._oracle_base='C:\app\ora_install_user'#ORACLE_BASE set from environment
orcl._pga_aggregate_target=587202560
orcl._sga_target=1090519040
orcl._shared_io_pool_size=0
orcl._shared_pool_size=251658240
orcl._streams_pool_size=0
*.audit_file_dest='C:\app\ora_install_user\admin\orcl\adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='C:\app\ora_install_user\oradata\orcl\control01.ctl','C:\app\ora_in
*.db_block_size=8192
*.db_domain='be.be'
*.db_name='orcl'
*.db_recovery_file_dest='C:\app\ora_install_user\fast_recovery_area'
*.db_recovery_file_dest_size=6930m
*.diagnostic_dest='C:\app\ora_install_user'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.enable_pluggable_database=true
*.memory_target=1600m
*.open_cursors=350
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

# Файл параметров

This PC ▶ Local Disk (C:) ▶ app ▶ ora\_install\_user ▶ product ▶ 12.1.0 ▶ dbhome\_1 ▶ database

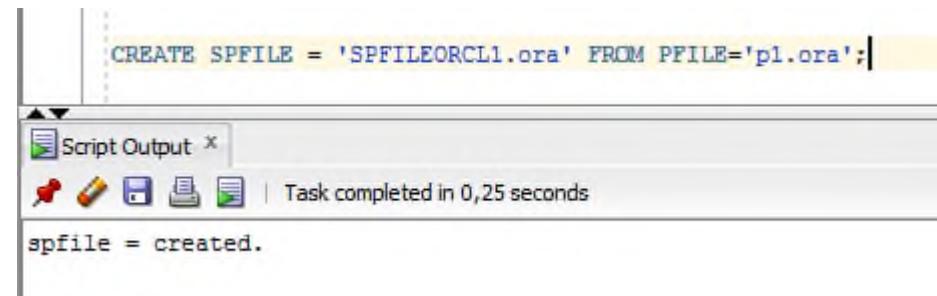
Name	Date modified	Type	Size
archive	29.08.2016 5:27	File folder	
hc_orcl.dat	29.08.2016 5:30	DAT File	2 KB
oradba	22.12.2005 4:07	Application	31 KB
oradim	26.09.2016 4:47	Text Document	1 KB
p1	27.09.2016 6:00	ORA File	2 KB
PWDorcl	29.08.2016 5:57	ORA File	8 KB
SPFILEORCL	27.09.2016 6:02	ORA File	4 KB



# Файл параметров

- ▶ Сформировать бинарный файл SPFILE параметров из текстового файла PFILE

```
*.db_block_size=8192
*.db_domain='be.by'
*.db_name='orcl'
*.db_recovery_file_dest='C:\app\ora_install_user\fast_recovery_a
*.db_recovery_file_dest_size=6930m
*.diagnostic_dest='C:\app\ora_install_user'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.enable_pluggable_database=true
*memory_target=1600m
*open_cursors=300
*processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```



```
SELECT name, value FROM V$PARAMETER where name = 'open_cursors';
```

Query Result

NAME	VALUE
open_cursors	300



# Файл параметров

## Порядок поиска имен

1. spfileORACLE\_SID.ora

2. spfile.ora

3. initORACLE\_SID.ora

### ▶ Переименован spfileorcl.ora

```
SQL> startup open;
ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file 'C:\APP\ORA_INSTALL_USER\PRODUCT\12.1.0
\DBHOME_1\DATABASE\INITORCL.ORA'
```

### ▶ Переименован p1.ora в initorcl.ora

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup open;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers   654311424 bytes
Redo Buffers        6873088 bytes
Database mounted.
Database opened.

SQL> show parameter spfile;
```

NAME	TYPE	VALUE
spfile	string	

# Файл параметров

- ▶ Переименован spfileorcl\_0.ora

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup open;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers   654311424 bytes
Redo Buffers        6873088 bytes

Database mounted.
Database opened.
SQL> show parameter spfile;

NAME                           TYPE        VALUE
spfile                         string      C:\APP\ORA_INSTALL_USER\PRODUC
                                T\12.1.0\DBHOME_1\DATABASE\SPF
                                ILEORCL.ORA

SQL>
```

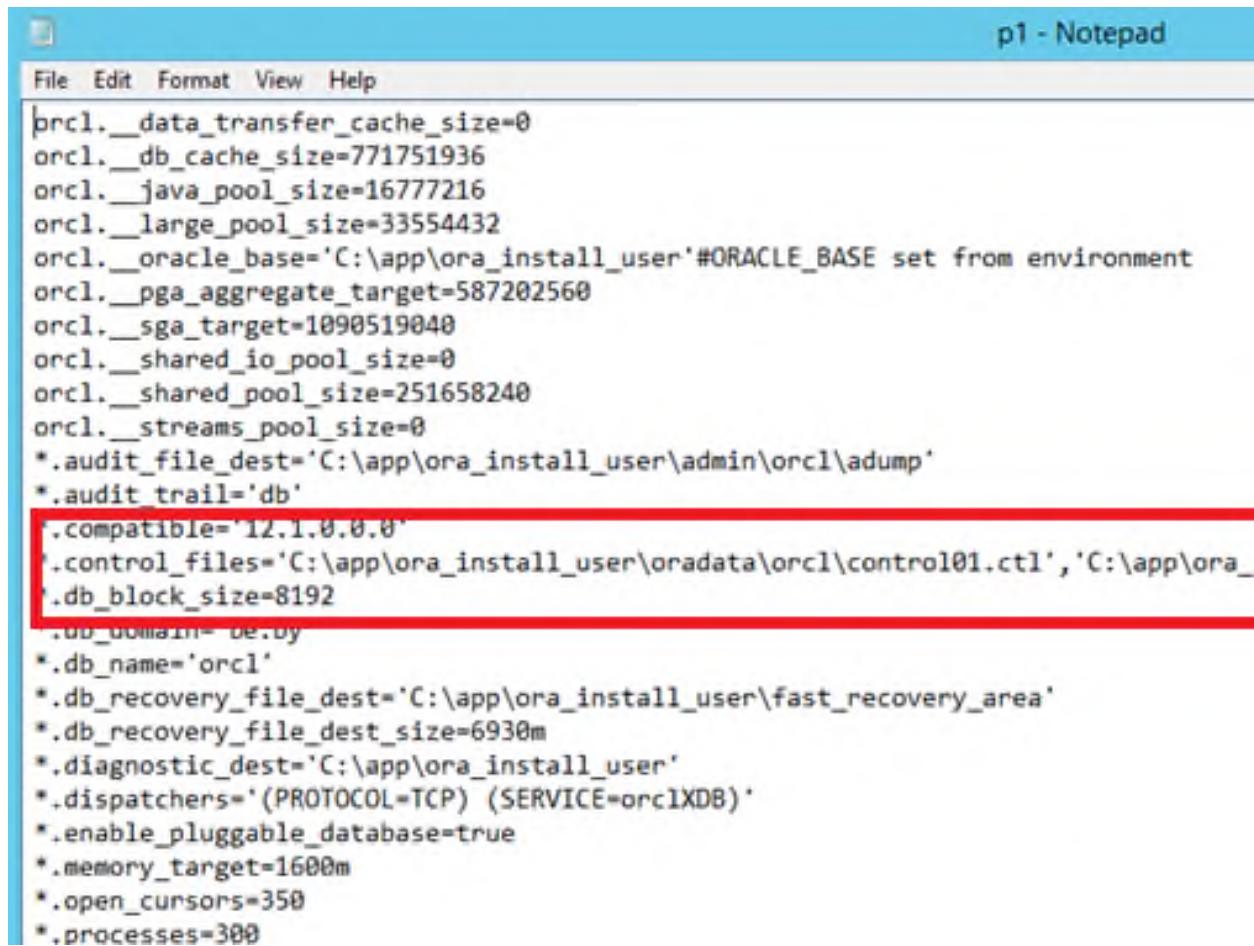
# Управляющие файлы

---

- ▶ **Control files** – файлы, содержащие имена (местоположение) основных физических файлов базы данных и некоторых параметров
- ▶ Используются для поиска других файлов операционной системы;
- ▶ Местоположение управляющих файлов экземпляр получает из файла параметров.
- ▶ По умолчанию для надежности создается 2 управляющих файла. Можно создать больше. Обычно их размещают на разных дисковых носителях (для надежности).



# Управляющие файлы



The screenshot shows a Windows Notepad window titled "p1 - Notepad". The window contains a list of Oracle database initialization parameters (spfile) in INI file format. The parameters include memory settings, log file locations, and connection details. A red rectangular box highlights several specific parameters: ".compatible", ".control\_files", ".db\_block\_size", ".db\_domain", ".db\_name", ".db\_recovery\_file\_dest", ".diagnostic\_dest", ".dispatchers", ".enable\_pluggable\_database", ".memory\_target", ".open\_cursors", and ".processes".

```
File Edit Format View Help
prcl._data_transfer_cache_size=0
orcl._db_cache_size=771751936
orcl._java_pool_size=16777216
orcl._large_pool_size=33554432
orcl._oracle_base='C:\app\ora_install_user'#ORACLE_BASE set from environment
orcl._pga_aggregate_target=587202560
orcl._sga_target=1090519040
orcl._shared_io_pool_size=0
orcl._shared_pool_size=251658240
orcl._streams_pool_size=0
*.audit_file_dest='C:\app\ora_install_user\admin\orcl\adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='C:\app\ora_install_user\oradata\orcl\control01.ctl','C:\app\ora_i
*.db_block_size=8192
*.db_domain= be.by
*.db_name='orcl'
*.db_recovery_file_dest='C:\app\ora_install_user\fast_recovery_area'
*.db_recovery_file_dest_size=6930m
*.diagnostic_dest='C:\app\ora_install_user'
*.dispatchers=(PROTOCOL=TCP) (SERVICE=orclXDB)
*.enable_pluggable_database=true
*.memory_target=1600m
*.open_cursors=350
*.processes=300
```

# Управляющие файлы

```
SHOW PARAMETER CONTROL;
```

Script Output X | Task completed in 0,265 seconds

NAME	TYPE	VALUE
control_file_record_keep_time	integer	7
control_files	string	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL01.CTL, C:
control_management_pack_access	string	DIAGNOSTIC+TUNING

```
SELECT NAME FROM V$CONTROLFILE;
```

Script Output X | Query Result X | All Rows Fetched: 2 in 0,078 seconds

NAME
1 C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL01.CTL
2 C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\CONTROL02.CTL

# Управляющие файлы

```
SELECT TYPE, RECORD_SIZE, RECORDS_TOTAL FROM V$CONTROLFILE_RECORD_SECTION;
```

Script Output x Query Result x

SQL | All Rows Fetched: 41 in 0,015 seconds

TYPE	RECORD_SIZE	RECORDS_TOTAL
1 DATABASE	316	1
2 CKPT PROGRESS	8180	11
3 REDO THREAD	256	8
4 REDO LOG	72	16
5 DATAFILE	520	1024
6 FILENAME	524	4146
7 TABLESPACE	68	1024
8 TEMPORARY FILENAME	56	1024
9 RMAN CONFIGURATION	1108	50
10 LOG HISTORY	56	292
11 OFFLINE RANGE	200	1063
12 ARCHIVED LOG	584	28
13 BACKUP SET	96	1022
14 BACKUP PIECE	780	1006
15 BACKUP DATAFILE	200	1063
16 BACKUP REDOLOG	76	215
17 DATAFILE COPY	736	1000
18 BACKUP CORRUPTION	44	1115
19 COPY CORRUPTION	40	1227
20 DELETED OBJECT	20	818

# Управляющие файлы – изменение

---

- ▶ Остановить Oracle (`shutdown transactional` или `immediate`);
- ▶ Скопировать один из управляющих файлов;
- ▶ Изменить параметр `CONTROL_FILES` в файле параметров;
- ▶ Стартовать Oracle (`startup open`).



# Управляющие файлы – изменение

```
SQL> startup open;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers    654311424 bytes
Redo Buffers        6873088 bytes
ORA-00214: control file 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL01.CTL'
version 20509 inconsistent with file
'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL03.CTL' version 20491

SQL> select name from v$controlfile;
no rows selected

SQL> show parameter control;

NAME                           TYPE        VALUE
control_file_record_keep_time integer     7
control_files                  string      C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL01.CTL, C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\CONTROL02.CTL, C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\CONTROL03.CTL
control_management_pack_access string      DIAGNOSTIC+TUNING
SQL> -
```

# Управляющие файлы – изменение

- ▶ Старт Oracle без одного из управляющих файлов;

```
SQL> startup open;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers   654311424 bytes
Redo Buffers        6873088 bytes
ORA-00205: error in identifying control file, check alert log for more info
```

# Управляющие файлы – изменение

- Если надо изменить управляющий файл, то следует создать сценарий, откорректировать его и выполнить



```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

database backup CONTROLFILE altered.
```

SQL> ORA\_ora\_19244@... Notepad

```
File Edit Format View Help
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS NOARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 1024
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\RED001.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\RED002.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\RED003.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 4 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\RED004.LOG'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBSEED\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\SYSAUD01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBSEED\SYSAUD01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\UNDOTBS01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\USERS01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBORCL\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBORCL\SYSAUD01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBORCL\SAMPLE_SCHEMA_USERS01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBORCL\EXAMPLE01.DBF',
  'C:\APP\TABLESPACES\TS_LEC.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDB_A\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDB_A\SYSAUD01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDB_A\USERS01.DBF'
```

# Файл паролей

- ▶ Предназначен для аутентификации администраторов базы данных

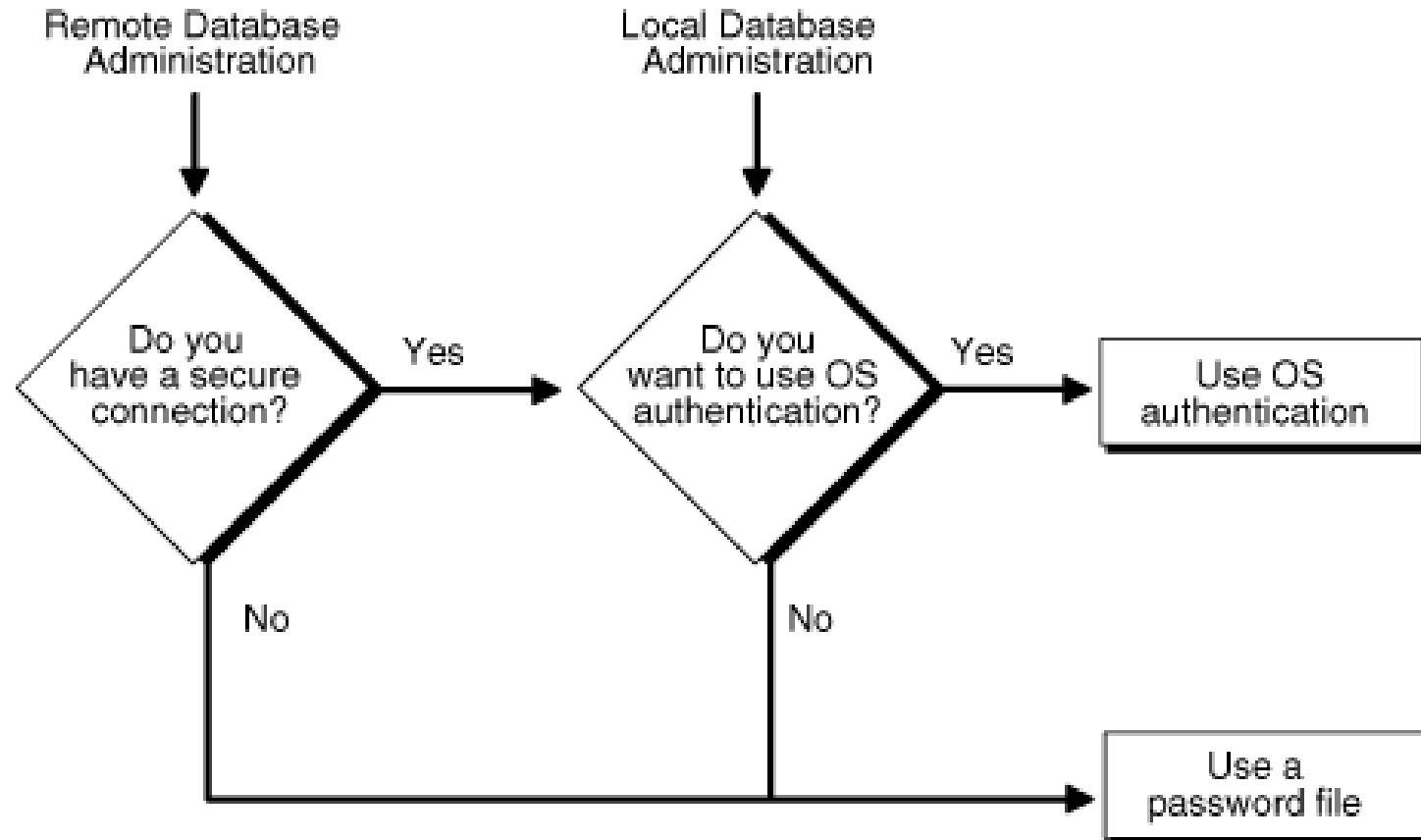
```
warning: You are no longer connected
SQL> connect /as sysdba;
Connected to an idle instance.
SQL>
```

- ▶ Можно создавать, пересоздавать и изменять

Platform	Required Name	Required Location
UNIX and Linux	orapwORACLE_SID	ORACLE_HOME/dbs
Windows	PWDORACLE_SID.ora	ORACLE_HOME\database



# Аутентификация администраторов



# Файл паролей

- ▶ Просмотр пользователей в файле паролей

SELECT * FROM V\$PFILE_USERS;								
	USERNAME	SYSDBA	SYSOPER	SYSASM	SYSBACKUP	SYSDG	SYSKM	CON_ID
1	SYS	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	0
2	SYSDG	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	1
3	SYSBACKUP	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	1
4	SYSKM	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	1



# Файл паролей

- ▶ Параметр использования файла пароля:
  - ▶ EXCLUSIVE
  - ▶ SHARED
  - ▶ NONE

```
show parameter remote_login_passwordfile;
```

NAME	TYPE	VALUE
remote_login_passwordfile	string	EXCLUSIVE

# Файлы сообщений

- ▶ Протоколы работы
- ▶ Трассировки
- ▶ Дампы

```
SELECT * FROM V$DIAG_INFO;
```

Query Result X

SQL | All Rows Fetched: 11 in 0,094 seconds

INST_ID	NAME	VALUE	CON_ID
1	1 Diag Enabled	TRUE	0
2	1 ADR Base	C:\APP\ORA_INSTALL_USER	0
3	1 ADR Home	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl	0
4	1 Diag Trace	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\trace	0
5	1 Diag Alert	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl>alert	0
6	1 Diag Incident	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\incident	0
7	1 Diag Cdump	C:\app\ora_install_user\diag\rdbms\orcl\orcl\cdump	0
8	1 Health Monitor	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\hm	0
9	1 Default Trace File	C:\APP\ORA_INSTALL_USER\diag\rdbms\orcl\orcl\trace\orcl_ora_144.trc	0
10	1 Active Problem Count	0	0
11	1 Active Incident Count	0	0

# Файлы сообщений

## ▶ Журнал ...\\ALERT\\LOG.XML

```
time='2016-09-27T05:30:04.193-07:00' org_id='oracle'
comp_id='rdbms'
msg_id='opixe:3293:2802784106' type='NOTIFICATION'
group='admin_ddl'
level='16' host_id='BOR' host_addr='fe80::cbf:3c5a:ab5a:dcfb%12'
module='SQL Developer' pid='1596'
<txt>Completed: alter database add logfile group 4 &apos;C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO04.LOG&apos;
size 50m blocksize 512
</txt>
</msg>
<msg time='2016-09-27T05:30:04.193-07:00' org_id='oracle'
comp_id='rdbms'
type='UNKNOWN' level='16' host_id='BOR'
host_addr='fe80::cbf:3c5a:ab5a:dcfb%12' module='SQL Developer'
pid='1596'>
<txt>ALTER SYSTEM SET open_cursors=350 SCOPE=SPFILE;
</txt>
</msg>
<msg time='2016-09-27T05:30:25.244-07:00' org_id='oracle'
comp_id='rdbms'
type='UNKNOWN' level='16' host_id='BOR'
host_addr='fe80::cbf:3c5a:ab5a:dcfb%12' module='SQL Developer'
pid='1596'>
<txt>ALTER SYSTEM SET open_cursors=350 SCOPE=SPFILE;
</txt>
</msg>
<msg time='2016-09-27T05:34:15.265-07:00' org_id='oracle'
comp_id='rdbms'
msg_id='opistp_real:1927:1141351617' type='NOTIFICATION'
group='shutdown'
```

# Журналы повтора

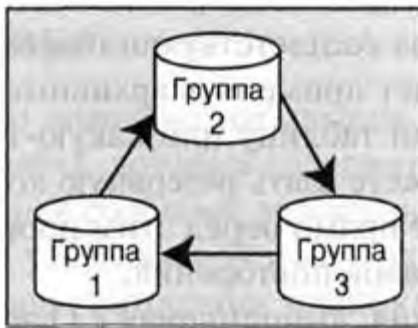
---

- ▶ **Журналы повторного выполнения** - дисковые ресурсы, в которых фиксируются изменения вносимых пользователями в базу данных;
- ▶ журнал представляет собой файл операционной системы;
- ▶ как минимум должно быть два файла;
- ▶ журналы применяются при восстановлении базы данных.



# Журналы повтора

- ▶ файлы журналов используются циклически (сначала запись идет в 1й файл, после заполнения во 2й, 3й..., затем снова в 1й)

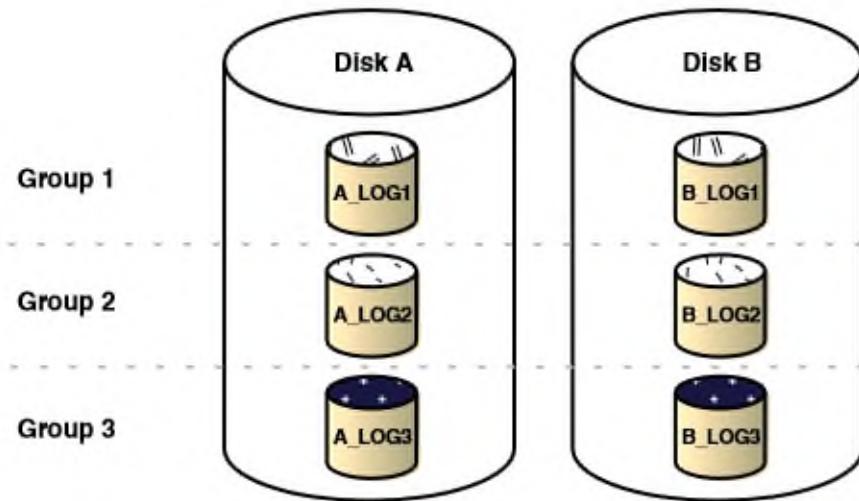


```
SELECT * FROM V$LOGFILE;
```

GROUP#	STATUS	TYPE	MEMBER	IS_RECOVERY_I
1	3 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO03.LOG	NO
2	2 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO02.LOG	NO
3	1 (null)	ONLINE	C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO01.LOG	NO

# Журналы повтора

- ▶ Мультиплексирование журналов повтора – поддержка несколько копий каждого журнала



```
SELECT GROUP#, STATUS, MEMBERS FROM V$LOG;
```

GROUP#	STATUS	MEMBERS
1	1 CURRENT	1
2	2 INACTIVE	1
3	3 INACTIVE	1

# Журналы повтора

- ▶ SCN – System change number – системный номер изменений в базе данных

SELECT \* FROM V\$LOG;

Query Result x

SQL | All Rows Fetched: 4 in 0,062 seconds

GROUP#	THREAD#	SEQUENCE#	BYTES	BLOCKSIZE	MEMBERS	ARCHIVED	STATUS	FIRST_CHANGE#	FIRST_TIME	...
1	1	1	121 52428800	512	1 NO		CURRENT	3832638	26.09.16	2
2	2	1	119 52428800	512	1 NO		INACTIVE	3797754	26.09.16	
3	3	1	120 52428800	512	1 NO		INACTIVE	3818813	26.09.16	

# Журналы повтора - добавление новой группы

```
ALTER DATABASE ADD LOGFILE GROUP 4 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO04.LOG'
SIZE 50m BLOCKSIZE 512;

select group#,
       sequence#,
       bytes,
       members,
       status,
       FIRST_CHANGE#
  from v$log;
```

Query Result x

SQL | All Rows Fetched: 4 in 0,046 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	121 52428800	1	CURRENT	3832638
2	2	119 52428800	1	INACTIVE	3797754
3	3	120 52428800	1	INACTIVE	3818813
4	4	0 52428800	1	UNUSED	0

# Журналы повтора - добавление файла в группу

```
ALTER DATABASE ADD LOGFILE MEMBER'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO041.LOG' TO GROUP 4 ;  
  
select group#,  
       sequence#,  
       bytes,  
       members,  
       status,  
       FIRST_CHANGE#  
  from v$log;
```

Script Output X    Query Result X

SQL | All Rows Fetched: 4 in 0,046 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	121 52428800	1	CURRENT	3832638
2	2	119 52428800	1	INACTIVE	3797754
3	3	120 52428800	1	INACTIVE	3818813
4	4	0 52428800	2	UNUSED	0

# Журналы повтора - переключение журналов

```
ALTER SYSTEM SWITCH LOGFILE;

SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0,062 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	121	52428800	1 ACTIVE	3832638
2	2	119	52428800	1 INACTIVE	3797754
3	3	120	52428800	1 INACTIVE	3818813
4	4	122	52428800	2 CURRENT	3864321

```
ALTER SYSTEM SWITCH LOGFILE;

SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0,078 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	121	52428800	1 ACTIVE	3832638
2	2	123	52428800	1 CURRENT	3864363
3	3	120	52428800	1 INACTIVE	3818813
4	4	122	52428800	2 ACTIVE	3864321

# Журналы повтора - переключение журналов

```
ALTER SYSTEM SWITCH LOGFILE;

SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0,047 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	52428800	1	ACTIVE	3832638
2	2	52428800	1	ACTIVE	3864363
3	3	52428800	1	CURRENT	3864438
4	4	52428800	2	ACTIVE	3864321

```
ALTER SYSTEM SWITCH LOGFILE;

SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

Script Output x Query Result x

SQL | All Rows Fetched: 4 in 0,047 seconds

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	52428800	1	CURRENT	3864455
2	2	52428800	1	ACTIVE	3864363
3	3	52428800	1	ACTIVE	3864438
4	4	52428800	2	ACTIVE	3864321

# Журналы повтора – удаление файла

- ▶ Группа должна быть не CURRENT
- ▶ Файл ОС не удаляется)

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a script editor containing two SQL statements:

```
ALTER DATABASE DROP LOGFILE MEMBER 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO041.LOG';
SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

In the bottom-right pane, the "Query Result" tab is selected, showing the output of the second query:

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
1	1	125	52428800	1 CURRENT	3864455
2	2	123	52428800	1 ACTIVE	3864363
3	3	124	52428800	1 ACTIVE	3864438
4	4	122	52428800	1 INACTIVE	3864321

The status column indicates that group 1 is CURRENT, while groups 2, 3, and 4 are ACTIVE or INACTIVE.

# Журналы повтора – удаление группы

```
ALTER DATABASE DROP LOGFILE GROUP 4;
SELECT GROUP#, SEQUENCE#,BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
```

Script Output    Query Result

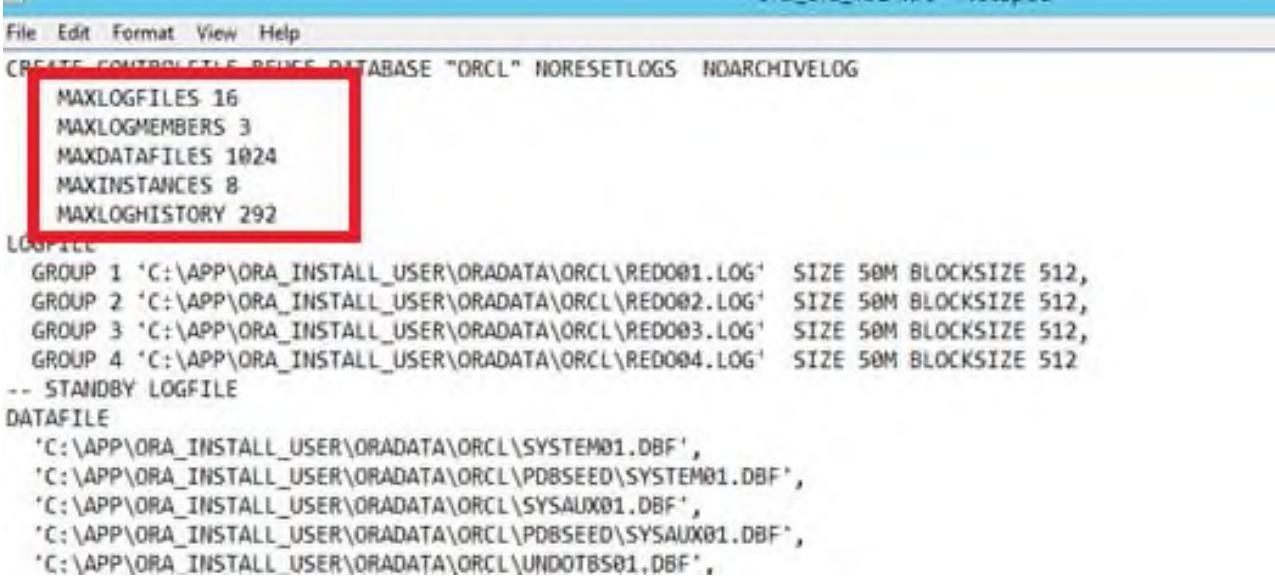
SQL | All Rows Fetched: 3 in 0,047 seconds

	GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS	FIRST_CHANGE#
	1	1	125	52428800	1 CURRENT	3864455
	2	2	123	52428800	1 INACTIVE	3864363
	3	3	124	52428800	1 INACTIVE	3864438

# Журналы повтора - параметры

Указывается в управляющих файлах:

- ▶ **MAXLOGFILES** - максимальное количество групп журналов повтора
- ▶ **MAXLOGMEMBERS** - максимальное количество файлов в группе



```
File Edit Format View Help
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS NOARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 1024
  MAXINSTANCES 8
  MAXLOGHISTORY 292
LOGFILE
  GROUP 1 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO01.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO02.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO03.LOG'  SIZE 50M BLOCKSIZE 512,
  GROUP 4 'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\REDO04.LOG'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBSEED\SYSTEM01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\SYSAUX01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\PDBSEED\SYSAUX01.DBF',
  'C:\APP\ORA_INSTALL_USER\ORADATA\ORCL\UNDOTBS01.DBF'.
```

# Архивы журналов повтора

- ▶ Режимы работы экземпляра:
  - ▶ ARCHIEVELOG
  - ▶ NOARCHIEVELOG

```
SELECT NAME, LOG_MODE FROM V$DATABASE;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0,032 seconds

NAME	LOG_MODE
1 ORCL	NOARCHIEVELOG

```
SELECT INSTANCE_NAME, ARCHIVER, ACTIVE_STATE FROM V$instance;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0,016 seconds

INSTANCE_NAME	ARCHIVER	ACTIVE_STATE
1 orcl	STOPPED	NORMAL



# Архивы

- ▶ Включить процесс архивирования
- ▶ Архивный файл появляется после переключения оперативного журнала.

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352 bytes
Variable Size       1006633960 bytes
Database Buffers   654311424 bytes
Redo Buffers        6873088 bytes

Database mounted.
SQL> alter database archivelog;

Database altered.

SQL> alter database open;

Database altered.

SQL> _
```

# Архивы

```
SELECT NAME, LOG_MODE FROM V$DATABASE;
```

Script Output × Query Result ×  
SQL | All Rows Fetched: 1 in 0,031 seconds

NAME	LOG_MODE
1 ORCL	ARCHIVELOG

```
SELECT INSTANCE_NAME, ARCHIVER, ACTIVE_STATE FROM V$instance;
```

Script Output × Query Result × Query Result 1 ×  
SQL | All Rows Fetched: 1 in 0,016 seconds

INSTANCE_NAME	ARCHIVER	ACTIVE_STATE
1 orcl	STARTED	NORMAL

# Архивы

```
SELECT * FROM V$LOG;
```

Script Output | Query Result | Query Result 1 | SQL | All Rows Fetched: 3 in 0,063 seconds

GROUP#	THREAD#	SEQUENCE#	BYTES	BLOCKSIZE	MEMBERS	ARCHIVED	STATUS	FIRST_CHANGE#	FIRST_TIME	NEXT
1	1	1	125	52428800	512	1 NO	CURRENT	3864455	27.09.16	28147
2	2	1	123	52428800	512	1 YES	INACTIVE	3864363	27.09.16	
3	3	1	124	52428800	512	1 YES	INACTIVE	3864438	27.09.16	

# Архивы журналов повтора

## ▶ После переключения журнала

```
ALTER SYSTEM SWITCH LOGFILE;
SELECT * FROM V$ARCHIVED_LOG;
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 1 in 0,063 seconds

RECID	STAMP	NAME
1	923650337	C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_125_CY07RKCR_.ARC

```
ALTER SYSTEM SWITCH LOGFILE;
SELECT * FROM V$ARCHIVED_LOG;
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 2 in 0 seconds

RECID	STAMP	NAME
1	923650337	C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_125_CY07RKCR_.ARC
2	923650483	C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_126_CY07X3M6_.ARC

# Архивы

```
ALTER SYSTEM SWITCH LOGFILE;
SELECT GROUP#, SEQUENCE#, BYTES, MEMBERS, STATUS, FIRST_CHANGE# FROM V$LOG;
SELECT Name, FIRST_CHANGE#, NEXT_CHANGE# FROM V$ARCHIVED_LOG;
```

GROUP #	SEQUENCE #	BYTES	MEMBERS	STATUS	FIRST_CHANGE #
1	1	128	52428800	1 CURRENT	3866172
2	2	126	52428800	1 INACTIVE	3866094
3	3	127	52428800	1 ACTIVE	3866147

NAME	FIRST_CHANGE #	NEXT_CHANGE #
1 C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_125_CY07RKCR_.ARC	3864455	3866094
2 C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_126_CY07X3M6_.ARC	3866094	3866147
3 C:\APP\ORA_INSTALL_USER\FAST_RECOVERY_AREA\ORCL\ARCHIVELOG\2016_09_27\01_MF_1_127_CY07YBDJ_.ARC	3866147	3866172

# Архивы

Worksheet | Query Builder

ARCHIVE LOG LIST;

Script Output | Task completed in 0,109 seconds

Database log mode Archive log Mode  
Automatic archival Enabled  
Archive destination USE\_DB\_RECOVERY\_FILE\_DEST  
Oldest online log sequence 126  
Next log sequence to archive 128  
Current log sequence 128

SHOW PARAMETER DB\_RECOVERY;

NAME	TYPE	VALUE
db_recovery_file_dest	string	C:\app\ora_install_user\fast_recovery_area
db_recovery_file_dest_size	big integer	6930M

PC ▶ Local Disk (C:) ▶ app ▶ ora\_install\_user ▶ fast\_recovery\_area ▶ orcl ▶ ARCHIVELOG ▶ 2016\_09\_27

Name	Date modified	Type	Size
01_MF_1_125_CYO7RKCR_.ARC	27.09.2016 9:32	ARC File	849 KB
01_MF_1_126_CYO7X3M6_.ARC	27.09.2016 9:34	ARC File	7 KB
01_MF_1_127_CYO7YBDJ_.ARC	27.09.2016 9:35	ARC File	13 KB

# Архивы

## Отключение процесса архивирования

```
SQL> startup mount;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size                  2403352 bytes
Variable Size              1006633960 bytes
Database Buffers           654311424 bytes
Redo Buffers                6873088 bytes
Database mounted.

SQL> alter database noarchivelog;

Database altered.
```

```
SQL> select name, log_mode from v$database;
NAME      LOG_MODE
----- -----
ORCL      NOARCHIELOG
```

The screenshot shows the Oracle SQL Developer interface. A query is being run:

```
SELECT INSTANCE_NAME, ARCHIVER, ACTIVE_STATE FROM V$instance;
```

The results are displayed in the "Query Result" tab:

INSTANCE_NAME	ARCHIVER	ACTIVE_STATE
1 orcl	STOPPED	NORMAL

Below the table, the status message "All Rows Fetched: 1 in 0,016 seconds" is visible.

# Вопросы?

---



# Администрирование баз данных и приложений

**Архитектура ORACLE: Области оперативной памяти**

Лекция 5

# При запуске экземпляра

---

- ▶ выделяется область разделяемой памяти, называемая SGA
  - ▶ запускаются фоновые процессы
- 
- ▶ Процесс - это механизм в операционной системе, который может выполнить последовательность шагов



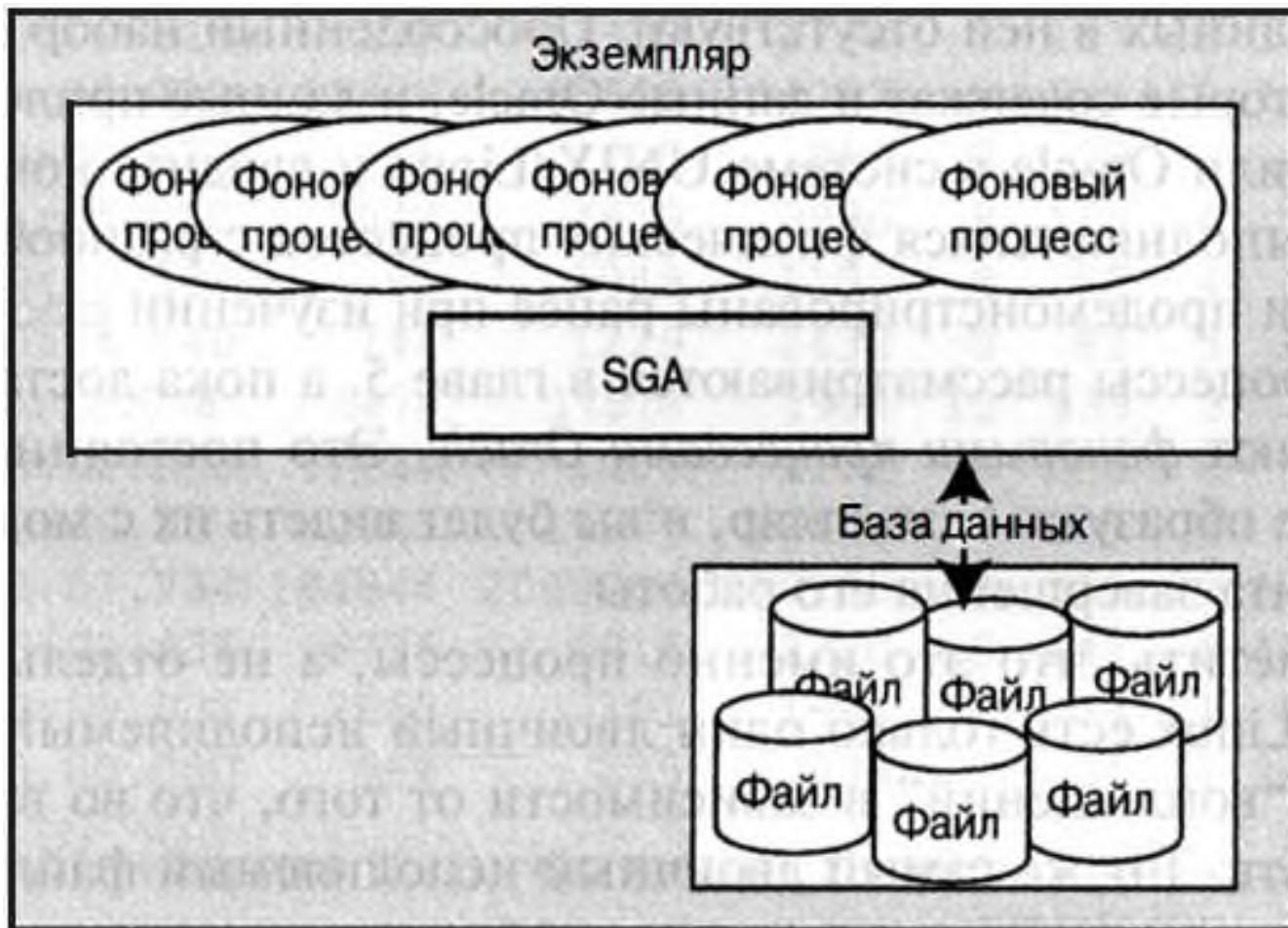
# Хранение и использование данных

---

- ▶ выполняемый код программы
- ▶ данные, которые совместно используются
- ▶ частные области данных для каждого подключенного пользователя



# Схема экземпляра



# Области оперативной памяти

---

- ▶ Системная Глобальная область - SGA
- ▶ Глобальные программные области - PGA



# SGA

---

- ▶ группа областей разделяемой памяти
- ▶ содержат данные и управляющую информацию для одного экземпляра Oracle
- ▶ совместно используется всеми серверными и фоновыми



# PGA

---

- ▶ создаваемая Oracle неразделяемая память при запуске серверного или фонового процесса
  - ▶ области памяти, которые содержат данные и управляющую информацию для серверного или фонового процесса, такие как пространство стека или информация сеанса
  - ▶ доступ к PGA является монопольным для серверного процесса
  - ▶ у каждого серверного процесса и фонового процесса есть собственная PGA
  - ▶ запись в PGA выполняется только кодом Oracle, действующим по запросу PGA
-

# PGA

---

- ▶ содержит стековое пространство
- ▶ содержит пользовательскую глобальную область (UGA)



# UGA

---

- ▶ Область курсоров
- ▶ Область хранения данных сеанса
- ▶ Рабочие области SQL:
  - ▶ Область сортировки для функций, таких как ORDER BY и GROUP BY
  - ▶ Область хеша для выполнения соединения таблиц
  - ▶ Область создания битового массива
  - ▶ Область слияния битовых массивов, используемую для того, чтобы разрешать план выполнения индекса битового массива



# UGA – User Global Area

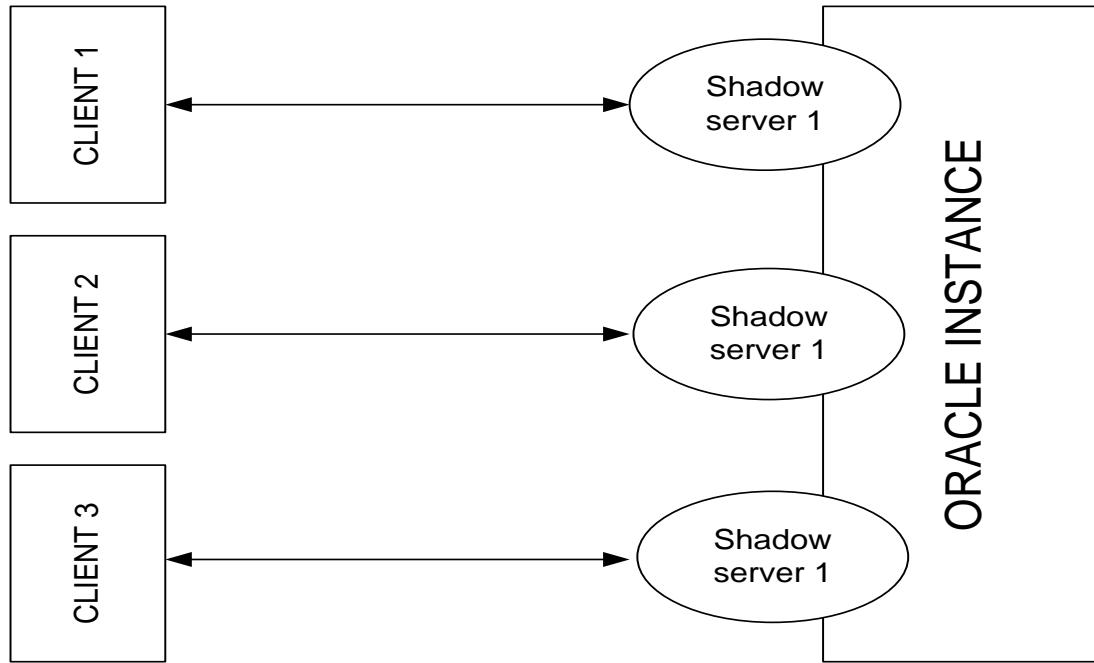
---

- ▶ Располагается в области SGA, если подключение выполнено посредством разделяемого сервера, в PGA остается только стековое пространство
- ▶ Располагается в области PGA, если подключение выполнено через выделенный сервер



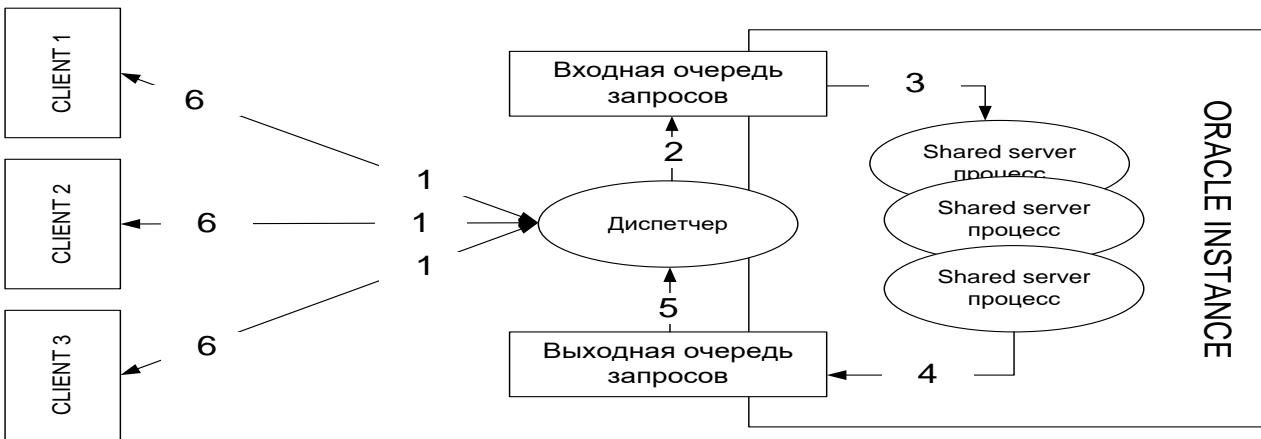
# Режим выделенного сервера

- ▶ **Dedicated server** (режим по умолчанию) – для каждого клиента выделяется отдельный выделенный серверный процесс (обработчик запросов, *dedicated server process*) который называется *shadow process* (теневой процесс).



# Режим разделяемого сервера

- ▶ **Shared server** – обрабатывает программа **dispatcher**:
- ▶ 1) получает запрос от клиента,
- ▶ 2) помещает их во входную очередь к разделяемым серверам;
- ▶ 3) незанятый разделяемый сервер извлекает и обрабатывает запрос;
- ▶ 4) после обработки разделяемый сервер помещает результат обработки в выходную очередь;
- ▶ 5) из очереди результат извлекает диспетчер;
- ▶ 6) диспетчер пересыпает результат клиенту



# Режимы соединений

```
select username, service_name, server from v$session where username is not null;
```

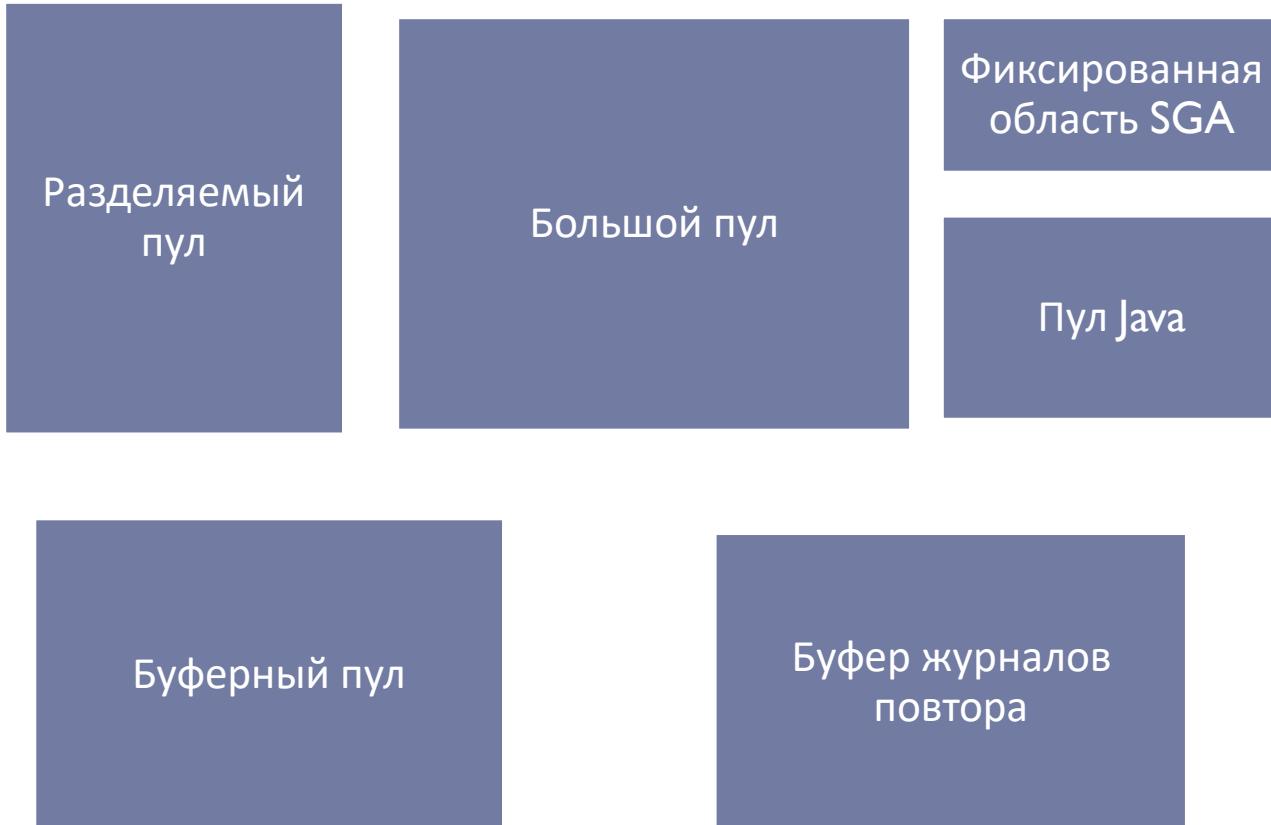
Query Result X

SQL | All Rows Fetched: 6 in 0 seconds

USERNAME	SERVICE_NAME	SERVER
1 SYS	pdb_a.be.by	DEDICATED
2 SYS	SYS\$USERS	DEDICATED
3 SYSTEM	pdb_a.be.by	DEDICATED
4 SYS	pdb_a.be.by	DEDICATED
5 C##USER_1	SYS\$USERS	DEDICATED
6 C##USER_1	pdb_a.be.by	DEDICATED

# Структура SGA

---



# Основные пулы области SGA

---

- ▶ Java pool
- ▶ Large pool
- ▶ Shared pool
- ▶ Streams pool
- ▶ “Null” pool



# Структура оперативной памяти Oracle 12c

```
SQL> startup open;
ORACLE instance started.

Total System Global Area 1670221824 bytes
Fixed Size          2403352  bytes
Variable Size       1006633960  bytes
Database Buffers   654311424  bytes
Redo Buffers        6873088  bytes
Database mounted.
Database opened.
```

Query Result x

SQL | All Rows Fetched: 4 in 0 seconds

NAME	VALUE	CON_ID
1 Fixed Size	2403352	0
2 Variable Size	1056965608	0
3 Database Buffers	603979776	0
4 Redo Buffers	6873088	0

Query Result x

SQL | All Rows Fetched: 1 in 0 seconds

SUM(VALUE)
1 1670221824

# Представления словаря

---

- ▶ v\$sga
- ▶ V\$sgastat
- ▶ v\$sga\_dynamic\_components
- ▶ v\$sga\_dynamic\_free\_memory



# Гранулы – granule

---

- ▶ Память различным пулам в SGA выделяется блоками, которые называются гранулами
- ▶ Одна гранула (granule) - это область памяти размером 4, 8 или 16 Мбайт
- ▶ Гранула является наименьшей единицей выделения памяти



# Представления словаря

```
select
    component,
    current_size,
    max_size,
    last_oper_mode,
    last_oper_time,
    granule_size,
    current_size/granule_size as Ratio
from v$sga_dynamic_components
where current_size > 0;
```

Query Result x

SQL | All Rows Fetched: 5 in 0 seconds

COMPONENT	CURRENT_SIZE	MAX_SIZE	LAST_OPER_MODE	LAST_OPER_TIME	GRANULE_SIZE	RATIO
1 shared pool	419430400	419430400	DEFERRED	04.10.16	16777216	25
2 large pool	33554432	150994944	DEFERRED	27.09.16	16777216	2
3 java pool	16777216	16777216	(null)	(null)	16777216	1
4 DEFAULT buffer cache	553648128	771751936	DEFERRED	04.10.16	16777216	33
5 Shared IO Pool	50331648	50331648	IMMEDIATE	27.09.16	16777216	3

# Представления словаря

```
select
sum(min_size),
sum(max_size),
sum(current_size)
from v$sga_dynamic_components;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,015 seconds

	SUM(MIN_SIZE)	SUM(MAX_SIZE)	SUM(CURRENT_SIZE)
1	855638016	1409286144	1073741824

```
select current_size from v$sga_dynamic_free_memory;
```

Query Result x

SQL | All Rows Fetched: 1 in 0 seconds

	CURRENT_SIZE
1	587202560

# Параметры SGA

- ▶ **SGA\_MAX\_SIZE** – указывает максимальный размер памяти

## SGA\_MAX\_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>SGA_MAX_SIZE = integer [K   M   G]</code>
Default value	Initial size of SGA at startup, dependent on the sizes of different pools in the SGA, such as buffer cache, shared pool, large pool, and so on.
Modifiable	No
Range of values	0 to operating system-dependent

# Параметры SGA

- ▶ **SGA\_TARGET** – указывает текущий (возможный) размер памяти

## SGA\_TARGET

Property	Description
Parameter type	Big integer
Syntax	<code>SGA_TARGET = integer [K   M   G]</code>
Default value	0 (SGA autotuning is disabled for DEFERRED mode autotuning requests, but allowed for IMMEDIATE mode autotuning requests)
Modifiable	ALTER SYSTEM
Range of values	64 MB to operating system-dependent
Basic	Yes

# Параметры SGA – изменение

```
SQL> alter system set sga_target=2G;
alter system set sga_target=2G
*
ERROR at line 1:
ORA-02097: parameter cannot be modified because specified value is invalid
ORA-00823: Specified value of sga_target greater than sga_max_size
```

```
SQL>
SQL>
SQL> show parameter sga
```

NAME	TYPE	VALUE
lock_sga	boolean	FALSE
pre_page_sga	boolean	TRUE
sga_max_size	big integer	1648M
sga_target	big integer	0
unified_audit_sga_queue_size	integer	1048576

```
SQL>
SQL> alter system set sga_target=128m;
```

```
System altered.
```

```
SQL> show parameter sga
```

NAME	TYPE	VALUE
lock_sga	boolean	FALSE
pre_page_sga	boolean	TRUE
sga_max_size	big integer	1648M
sga_target	big integer	128M
unified_audit_sga_queue_size	integer	1048576

```
select sum(bytes) from v$sgastat;
SUM(BYTES)
1267342504
```

# Структура SGA

---



# Буферный пул

---

- ▶ Буферный пул (буферный кэш) – область SGA, которая содержит образы блоков, считанные из файлов данных или созданные динамически, чтобы реализовать модель согласованного чтения
- ▶ Совместно используется всеми пользователями



# Буферный пул

---

- ▶ Пользовательский процесс БД требует определенный фрагмент данных
- ▶ Поиск данных в буферном пуле
- ▶ Если данные обнаружены, то их можно считать прямо из памяти
- ▶ Если данные не обнаружены, то придется скопировать блок данных из файла данных на диске в буфер, чтобы получить доступ к данным



# Буферный пул

---

- ▶ Буферный пул (буферный кэш) хранит блоки данных табличных пространств
- ▶ Блок – единица обмена информацией между оперативной памятью и диском
- ▶ С каждым блоком связан счетчик использования
- ▶ Списки блоков упорядочены по количеству обращений к блоку



# Буферный пул

---

- ▶ Поддерживается два списка блоков:
  - ▶ список грязных блоков (отличаются от своей копии на диске и должны быть записаны в табличное пространство)
  - ▶ список чистых блоков (не измененные блоки)



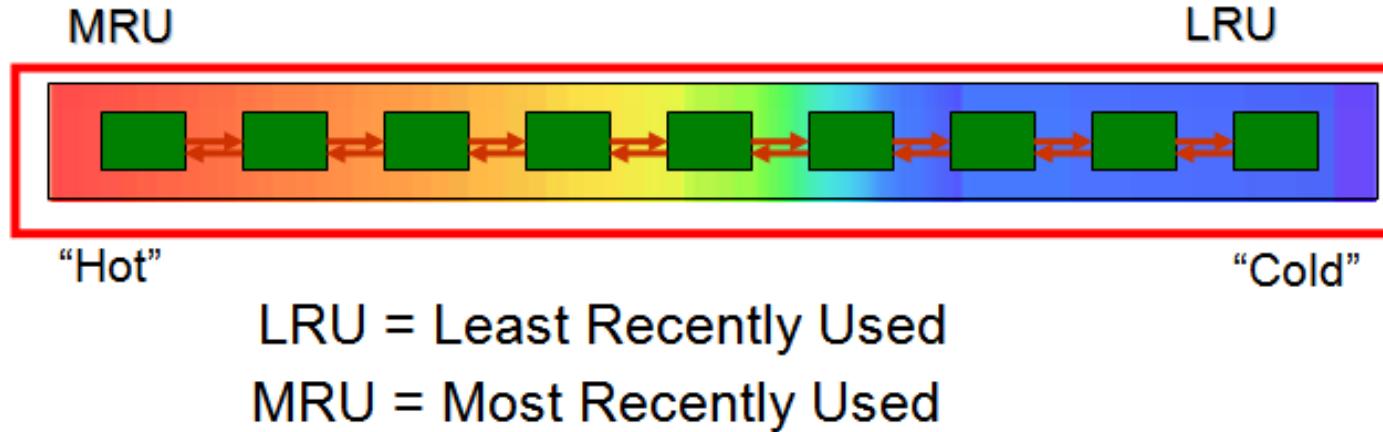
# Буферный пул

---

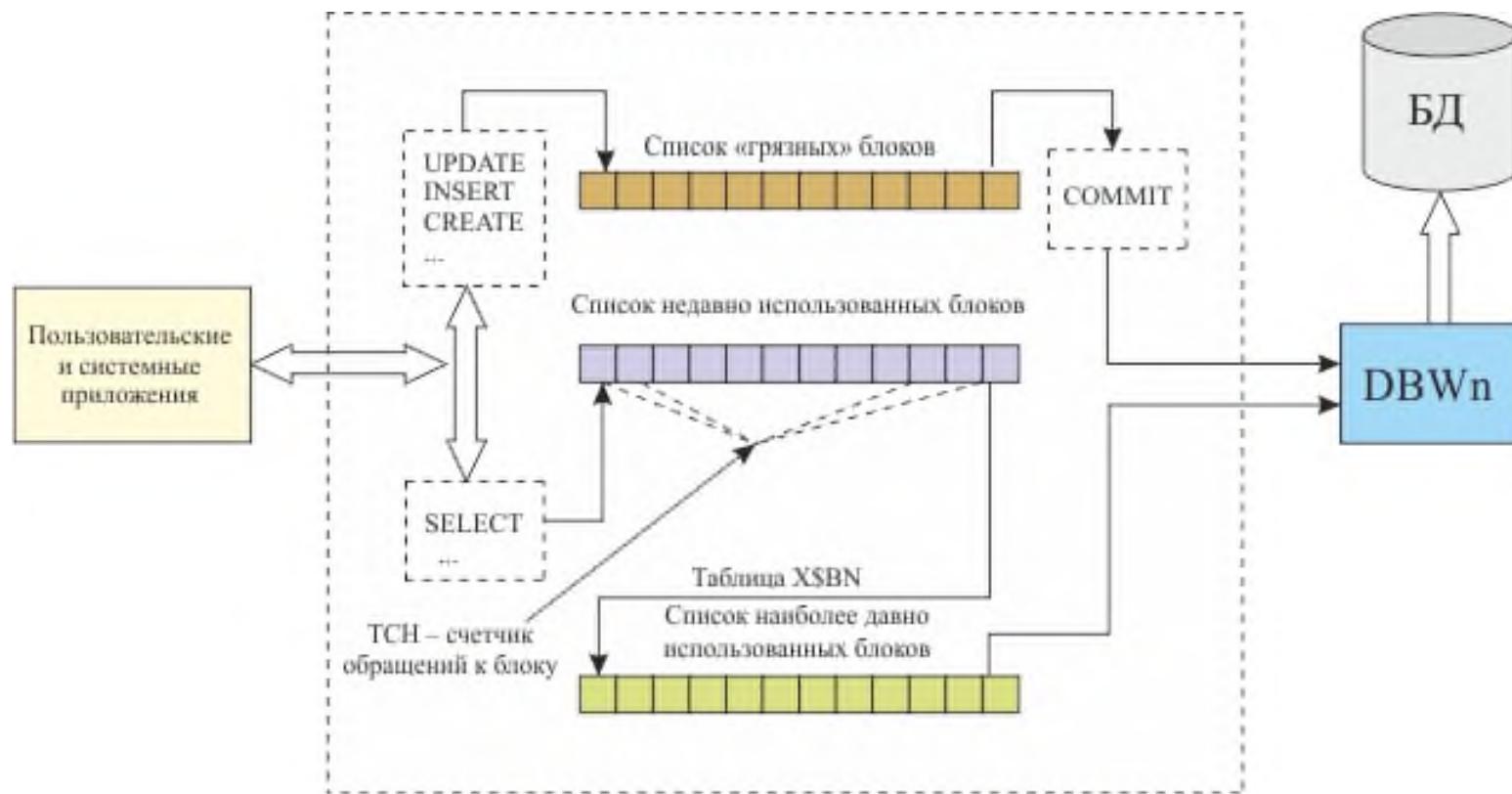
- ▶ Алгоритм LRU (least recently used) – первыми вытесняются блоки с наименьшим значением счетчика
- ▶ Запись грязных блоков на диск осуществляется в 4х случаях:
  - ▶ 1) истечение тайм-аута (3 сек);
  - ▶ 2) контрольная точка;
  - ▶ 3) превышение длины грязных блоков заданного лимита;
  - ▶ 4) процесс не может обнаружить свободный блок.



# Буферный пул – вытеснение блоков



# Буферный пул – вытеснение блоков



# Пулы буферного кэша

- ▶ DEFAULT
- ▶ KEEP
- ▶ RECYCLE

```
select
component,
min_size,
current_size
from v$sga_dynamic_components;
```

Query Result | All Rows Fetched: 15 in 0 seconds

COMPONENT	MIN_SIZE	CURRENT_SIZE
1 shared pool	251658240	419430400
2 large pool	33554432	33554432
3 java pool	16777216	16777216
4 streams pool	0	0
5 DEFAULT buffer cache	553648128	553648128
6 KEEP buffer cache	0	0
7 RECYCLE buffer cache	0	0
8 DEFAULT 2K buffer cache	0	0
9 DEFAULT 4K buffer cache	0	0
10 DEFAULT 8K buffer cache	0	0
11 DEFAULT 16K buffer cache	0	0
12 DEFAULT 32K buffer cache	0	0
13 Shared IO Pool	0	50331648

# Параметры пулов буферного кэша

- ▶ Устанавливаются  
`alter system set ...`
- ▶ В файле параметров

## DB\_KEEP\_CACHE\_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_KEEP_CACHE_SIZE = integer [K   M   G]</code>
Default value	0 ( <code>DB_KEEP_CACHE_SIZE</code> is not configured by default)
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the granule size * number of processor groups, or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

## DB\_CACHE\_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_CACHE_SIZE = integer [K   M   G]</code>
Default value	If <code>SGA_TARGET</code> is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If <code>SGA_TARGET</code> is not set, then the default is either 48 MB or 4 MB * number of CPUs, whichever is greater
Modifiable	ALTER SYSTEM
Basic	No

## DB\_RECYCLE\_CACHE\_SIZE

Property	Description
Parameter type	Big integer
Syntax	<code>DB_RECYCLE_CACHE_SIZE = integer [K   M   G]</code>
Default value	0 ( <code>DB_RECYCLE_CACHE_SIZE</code> is not configured by default)
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are automatically modified to be either the granule size * number of processor groups, or 4 MB * number of CPUs, whichever is greater) Maximum: operating system-dependent
Basic	No

# Пулы буферного кэша

---

- ▶ DEFAULT
- ▶ KEEP
- ▶ RECYCLE

```
alter system set db_cache_size=600m scope=spfile;
alter system set db_keep_cache_size=100m scope=spfile;
alter system set db_recycle_cache_size=100m scope=spfile;
```



# Пулы буферного кэша

```
select component, min_size, current_size, max_size
from v$sga_dynamic_components where component like '%cache%' ;
```

COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE
DEFAULT buffer cache	637534208	754974720	771751936
KEEP buffer cache	117440512	117440512	117440512
RECYCLE buffer cache	117440512	117440512	117440512
DEFAULT 2K buffer cache	0	0	0
DEFAULT 4K buffer cache	0	0	0
DEFAULT 8K buffer cache	0	0	0
DEFAULT 16K buffer cache	0	0	0
DEFAULT 32K buffer cache	0	0	0

```
select name, resize_state, block_size, buffers, prev_buffers from v$buffer_pool
```

NAME	RESIZE_STATE	BLOCK_SIZE	BUFFERS	PREV_BUFFERS
KEEP	STATIC	8192	13769	0
RECYCLE	STATIC	8192	13769	0
DEFAULT	STATIC	8192	80546	86548

# Пулы буферного кэша

```
create pfile='XXX.ora' from spfile;
```

```
XXX.ora — Блокнот
Файл Правка Формат Вид Справка
isus.__data_transfer_cache_size=0
isus.__db_cache_size=687865856
isus.__java_pool_size=16777216
isus.__large_pool_size=33554432
isus.__oracle_base='c:\app\oracle'#ORACLE_BASE set from environment
isus.__pga_aggregate_target=486539264
isus.__sga_target=1241513984
isus.__shared_io_pool_size=50331648
isus.__shared_pool_size=201326592
isus.__streams_pool_size=0
*.audit_file_dest='C:\app\oracle\admin\ISUS\adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.control_files='C:\app\oracle\oradata\ISUS\control01.ctl','C:\ap
*.db_block_size=8192
*.db_cache_size=629145600
*.db_domain=''
*.db_keep_cache_size=104857600
*.db_name='ISUS'
*.db_recovery_file_dest='C:\app\oracle\fast_recovery_area'
*.db_recovery_file_dest_size=6930m
*.db_recycle_cache_size=104857600
*.diagnostic_dest='C:\app\oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ISUSXDB)'
*.enable_pluggable_database=true
*.memory_target=1638m
*.open_cursors=350
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

# Помещение таблицы в определенный пул

```
create table XXX (k int) storage(buffer_pool keep)    tablespace users;
create table YYY (k int) storage(buffer_pool recycle) tablespace users;
create table zzz (k int) storage(buffer_pool default) tablespace users; -- по умолчанию

insert into XXX values(1);
insert into YYY values(1);
insert into ZZZ values(1);
commit;

select segment_name, segment_type,tablespace_name, buffer_pool from user_segments
where segment_name in('XXX','YYY','ZZZ');
```

SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BUFFER_POOL
XXX	TABLE	USERS	KEEP
YYY	TABLE	USERS	RECYCLE
ZZZ	TABLE	USERS	DEFAULT

# CACHE

CACHE – помещение таблицы в конец LRU-списка (для малых таблиц) обычно в default pool

```
create table CCC  (k int) cache    tablespace users;
create table CCC1 (k int) cache     storage(buffer_pool keep) tablespace users;
create table CCC2 (k int) cache     storage(buffer_pool recycle) tablespace users;

insert into CCC values(1);
insert into CCC1 values(1);
insert into CCC2 values(1);
commit;

select segment_name, segment_type,tablespace_name, buffer_pool from user_segments
where segment_name in('CCC', 'CCC1', 'CCC2');
```

SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	BUFFER_POOL
CCC	TABLE	USERS	DEFAULT
CCC1	TABLE	USERS	KEEP
CCC2	TABLE	USERS	RECYCLE

# Структура SGA

---



# Буфер журналов повторного выполнения

---

- ▶ Буфер журнала повторного выполнения предназначен для временного циклического хранения данных журнала повтора
- ▶ Позволяет ускорить работу сервера за счет буферизации
- ▶ Содержимое сбрасывается на диск (в журнал повтора) в 4-х случаях:
  - ▶ 1) каждые три секунды;
  - ▶ 2) при фиксации транзакции;
  - ▶ 3) при заполнении буфера на 1/3;
  - ▶ 4) если в буфере более 1м данных журнала повтора.



# Параметр LOG\_BUFFER

## LOG\_BUFFER

Property	Description
Parameter type	Integer
Default value	5 MB to 32 MB, depending on the size of the SGA, CPU count, and whether the operating system is 32-bit or 64-bit
Modifiable	No
Range of values	2 MB to operating system-dependent
Basic	No

```
show parameter log_buffer
```

NAME	TYPE	VALUE
log_buffer	integer	6344704



# Ожидание освобождения буфера журналов повтора

- ▶ Redo buffer allocation retries – статистика – количество случаев ожидания процессами освобождения буфера журнала повтора:
  - ▶ буфер занят
  - ▶ процесс LGWR не сбросил данные на диск
  - ▶ процесс ждет освобождения буфера для записи информации об изменении базы данных

Значение должно быть равно 0, иначе надо увеличивать размер буфера

```
select name, value from v$sysstat where name like '%redo%retries'  


| NAME                           | VALUE |
|--------------------------------|-------|
| redo buffer allocation retries | 0     |


```

# Структура SGA

---



## Фиксированная область SGA

---

- ▶ хранит переменные, указывающие на другие области памяти, значения параметров;
- ▶ представляет собой загрузочный бинарный код;
- ▶ размер области зависит от платформы, версии операционной системы;
- ▶ размером фиксированной области SGA управлять нельзя.



# Структура SGA

---



# Разделяемый пул

---

- ▶ Библиотечный кэш
- ▶ Разделяемую область SQL
- ▶ Кэш словаря данных
- ▶ Управляющие структуры



# Поддержка словаря данных

---

- ▶ Кэш словаря данных – содержит данные в виде строк
  - ▶ Библиотечный кэш
- 
- ▶ Все пользовательские процессы Oracle совместно используют эти два кэша для доступа к информации о словаре данных



# Поддержка словаря данных

---

- ▶ Каждый SQL-оператор распознается
- ▶ Для нового SQL-оператора выделяется память в разделяемом пуле
- ▶ Когда два пользователя выполняют одно и то же предложение SQL, оператор используется повторно из разделяемой области SQL
- ▶ Аналогично обрабатывается программные блоки



# Разделяемый пул

```
select
  component,
  min_size,
  current_size,
  max_size
  from v$sga_dynamic_components
 where component = 'shared pool';
```

Query Result X

SQL | All Rows Fetched: 1 in 0 seconds

	COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE	
1	shared pool	251658240	419430400	419430400	

# Управление размером разделяемого пула

- ▶ Устанавливаются alter system set ...
- ▶ В файле параметров

```
SQL> show parameter shared_pool
NAME          TYPE        VALUE
shared_pool_reserved_size  big integer 9227468
shared_pool_size          big integer 0
SQL> -
```

## SHARED\_POOL\_SIZE

Property	Description
Parameter type	Big integer
Syntax	SHARED_POOL_SIZE = integer [K   M   G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set (32-bit platforms): 64 MB, rounded up to the nearest granule size.  If SGA_TARGET is not set (64-bit platforms): 128 MB, rounded up to the nearest granule size.
For considerations when dealing with database instances using ASM, see "SHARED_POOL_SIZE and Automatic Storage Management".	
Modifiable	ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent

```
SQL>
SQL> alter system set shared_pool_size=180m;
System altered.
```

```
SQL> show parameter shared_pool
```

```
NAME          TYPE        VALUE
shared_pool_reserved_size  big integer 9227468
shared_pool_size          big integer 192M
SQL>
```

# Управление размером разделяемого пула

```
SQL> alter system set shared_pool_size=208m;
```

```
System altered.
```

```
SQL> show parameter shared_pool
```

NAME	TYPE	VALUE
shared_pool_reserved_size	big integer	9227468
shared_pool_size	big integer	208M

```
SQL> _
```

```
select component, min_size, current_size, max_size  
from v$sga_dynamic_components where component = 'shared pool' ;
```

COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE
shared pool	167772160	218103808	218103808



# Объекты в разделяемом пуле

```
select pool, name, bytes from v$sgastat where pool = 'shared pool';
```

POOL	NAME	BYTES
1	shared pool db_files	453448
2	shared pool KCFIS SGA	8
3	shared pool KTC txn rsrc cnt	88
4	shared pool os statistics	13096
5	shared pool kdlxdup swapp	24
6	shared pool KGLH0	10407544
7	shared pool KQF optimizer stats table	2224
8	shared pool namrec_kfdsg	24
9	shared pool object temp hash table la	163840
.0	shared pool keomm:emptyMon_keommSg	7168
.1	shared pool object level stat table	1056
.2	shared pool kgnfs gid list	104
.3	shared pool HM_MESSAGE	1504
.4	shared pool PQ/ResSched	32784
.5	shared pool generic process shared st	16
.6	shared pool KTSJ state object	1408
.7	shared pool SGA structure for kelr	2456
.8	shared pool KKJ SHRD WRQS	240
.9	shared pool zafwctx	40608
.10	shared pool z1llab Group Tree Heap De	160
.11	shared pool wcrn oca	160

```
select count(*) from v$sgastat  
where pool = 'shared pool';
```

Query Result	
	SQL
1	1201

# Структура SGA

---



# Большой пул

---

- ▶ Большой пул - область памяти SGA, применяемая для хранения больших фрагментов памяти
- ▶ В этой области не применяется вытеснение по алгоритму LRU
- ▶ память становится свободной сразу после того, как перестает использоваться
- ▶ аналог RECYCLE (разделяемый пул – KEEP)
- ▶ хранятся данные при резервном копировании (RMAN), специальные области UGA и пр.



# Большой пул

```
select
pool,
name,
bytes
from v$sgastat
where pool = 'large pool';
```

Query Result x

All Rows Fetched: 10 in 0 seconds

POOL	NAME	BYTES
1 large pool	PX msg pool	491520
2 large pool	SWRF Metric Eidbuf	368640
3 large pool	SWRF Metric CHBs	1802240
4 large pool	SWRF Metric Eidbuf	368640
5 large pool	SWRF Metric CHBs	1802240
6 large pool	SWRF Metric CHBs	1802240
7 large pool	free memory	24379392
8 large pool	SWRF Metric CHBs	1802240
9 large pool	SWRF Metric Eidbuf	368640
10 large pool	SWRF Metric Eidbuf	368640

```
select
component,
min_size,
current_size,
max_size
from v$sga_dynamic_components
where component = 'large pool';
```

Query Result x

All Rows Fetched: 1 in 0 seconds

COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE
1 large pool	33554432	33554432	150994944

# Большой пул – параметры

## LARGE\_POOL\_SIZE

Property	Description
Parameter type	Big integer
Syntax	LARGE_POOL_SIZE = <i>integer</i> [K   M   G]
Default value	If SGA_TARGET is set, but a value is not specified for LARGE_POOL_SIZE, then the default is 0 (internally determined by the Oracle database). If LARGE_POOL_SIZE is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set, 0 if both of the following are true: <ul style="list-style-type: none"><li>• The pool is not required by parallel execution</li><li>• DBWR_IO_SLAVES is not set</li></ul> Otherwise, derived from the values of PARALLEL_MAX_SERVERS, PARALLEL_THREADS_PER_CPU, CLUSTER_DATABASE_INSTANCES, DISPATCHERS, and DBWR_IO_SLAVES.
Modifiable	ALTER SYSTEM
Range of values	Minimum: the granule size Maximum: operating system-dependent
Basic	No

```
show parameter large_pool
```

NAME	TYPE	VALUE
large_pool_size	big integer	0



# Большой пул – параметры

```
SQL>
SQL>
SQL> alter system set large_pool_size=2m;
System altered.

SQL> show parameter large_pool
NAME                      TYPE         VALUE
large_pool_size            big integer 16M
SQL> alter system set large_pool_size=18m;
System altered.

SQL> show parameter large_pool
NAME                      TYPE         VALUE
large_pool_size            big integer 32M
SQL> _
```

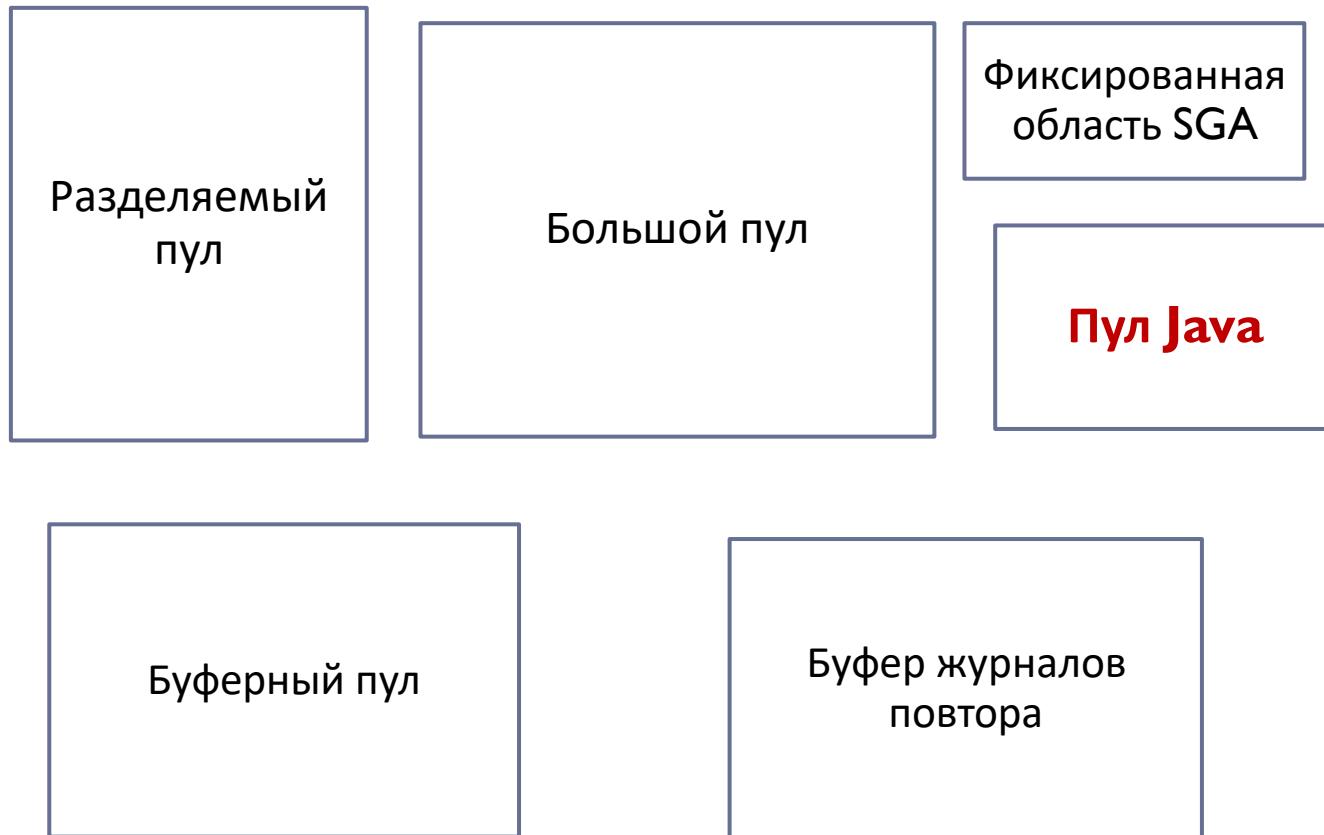
```
select component, min_size, current_size, max_size
from v$sga_dynamic_components where component = 'large pool' ;

```

COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE
large pool	33554432	33554432	150994944

# Структура SGA

---



# Java-пул

- ▶ Java-пул предназначен для работы Java-машины;

```
select
component,
min_size,
current_size,
max_size
from v$sga_dynamic_components
where component = 'java pool';
```

Query Result x

All Rows Fetched: 1 in 0 seconds

COMPONENT	MIN_SIZE	CURRENT_SIZE	MAX_SIZE
1 java pool	16777216	16777216	16777216

```
select
pool,
name,
bytes
from v$sgastat
where pool = 'java pool';
```

Query Result x

All Rows Fetched: 1 in 0 seconds

POOL	NAME	BYTES
1 java pool	free memory	16777216

# Параметры Java-пула

## JAVA\_POOL\_SIZE

Property	Description
Parameter type	Big integer
Syntax	JAVA_POOL_SIZE = <i>integer</i> [K   M   G]
Default value	If SGA_TARGET is set: If the parameter is not specified, then the default is 0 (internally determined by the Oracle Database). If the parameter is specified, then the user-specified value indicates a minimum value for the memory pool. If SGA_TARGET is not set: 24 MB, rounded up to the nearest granule size
Modifiable	ALTER SYSTEM
Range of values	Minimum: 0 (values greater than zero are rounded up to the nearest granule size) Maximum: operating system-dependent
Basic	No

```
SQL> alter system set java_pool_size=18m;
SQL> show parameter java_pool
NAME          TYPE        VALUE
java_pool_size big integer 32M
SQL>
SQL>
```

```
show parameter java_pool_size
NAME          TYPE        VALUE
java_pool_size big integer 0
```

# Вопросы?

---

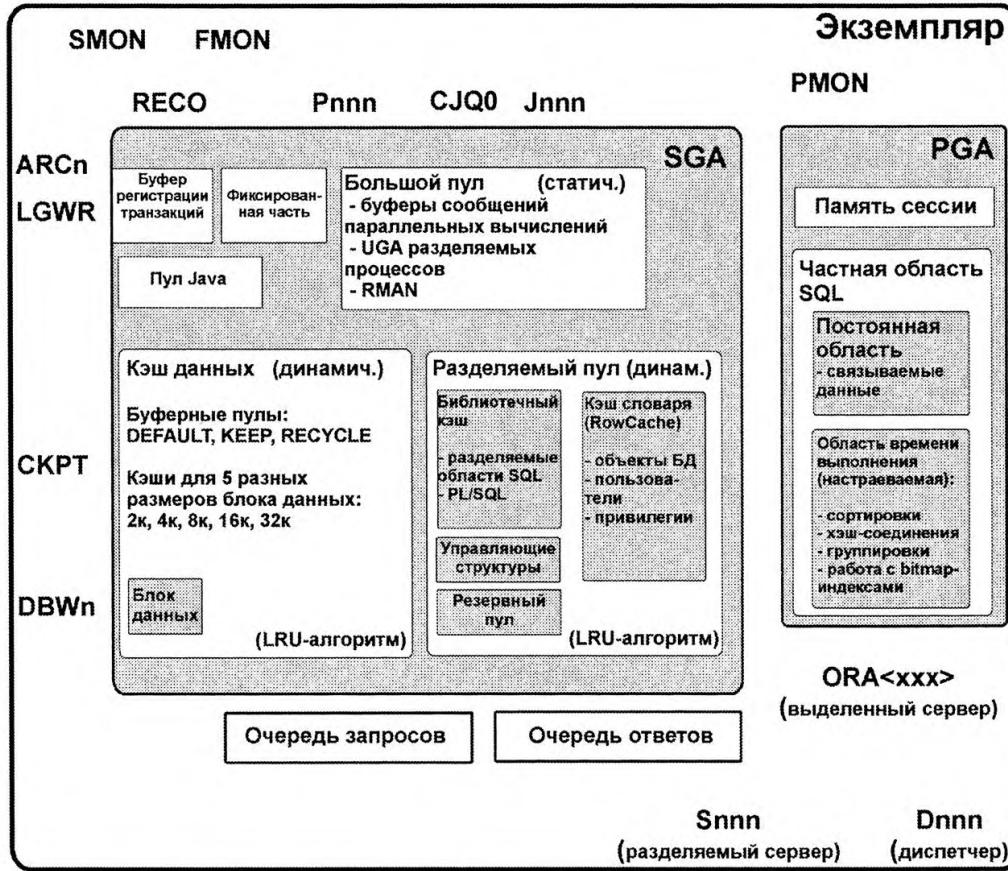


# **Архитектура ORACLE**

**Процессы**

**Лекция 6**

# Архитектура Oracle



# Процессы и потоки Windows

- ▶ Процесс (process) – механизм ОС Windows, осуществляющий запуск и выполнение приложений
  - ▶ Процесс создается, когда запускается приложение
  - ▶ В общем случае выполняется в собственной области памяти
- ▶ Поток (thread) – индивидуальная ветвь внутри процесса, выполняющая конкретные программные инструкции

▫	Oracle MTS Recovery Service OracleOraDB12Home1MTSRecoveryService	0%	2,2 MB
▫	Oracle RDBMS Kernel Executable OracleServiceORCL	0%	1 223,6 MB
▫	Oracle TNSLNR Executable OracleOraDB12Home1TNSListener	0%	4,5 MB
▫	oravssw OracleVssWriterORCL	0%	1,8 MB



# Процессы Oracle

---

- ▶ Серверные процессы – процессы, выполняющиеся на основании клиентского запроса
- ▶ Фоновые процессы – запускаются вместе с базой данных и выполняют разнообразные задачи обслуживания
- ▶ Подчиненные процессы – аналогичны фоновым, но выполняют дополнительные действия для фонового или серверного процессов



# Серверные процессы

---

- ▶ Клиентский процесс прямо взаимодействует с соответствующим серверным процессом по сети
- ▶ Первичное установление соединения происходит через специальный серверный процесс Oracle Net Listener
- ▶ Серверный процесс:
  - ▶ Получает и выполняет SQL-операторы
  - ▶ Читает файлы данных
  - ▶ Осуществляет поиск в кеше
- ▶ Назначение — отвечать на получаемые SQL-запросы
- ▶ Клиентом может быть любая программа, пытающаяся подключиться к СУБД



# Серверные процессы

---

- ▶ Отправка SQL-запроса `select * from students;`
  - ▶ Производит синтаксический разбор
  - ▶ Помещает в разделяемый пул (находит в пуле)
  - ▶ Создает план запроса и выполняет его
  - ▶ При необходимости, производит поиск данных в буферном кэше или запрашивает в кэш с диска



# Oracle Net Listener

---

- ▶ Oracle Net Listener – процесс на стороне сервера, прослушивающий входящие запросы клиента на соединение с экземпляром.
- ▶ Listener – это программа-сервер, прослушивающая TCP-порт, принимающая запросы на соединение с Oracle экземпляром от программ-клиентов.
- ▶ В результате успешной работы Listener устанавливается соединение между программой-клиентом и обработчиком запросов экземпляра.
- ▶ По умолчанию TCP-порт 1521



# Oracle Net Listener – Службы

- D	▷ Oracle MTS Recovery Service	0%	2,2 MB
- D	▷ Oracle RDBMS Kernel Executable	0%	1 224,3 MB
- U	▷ Oracle TNSLNR Executable OracleOraDB12Home1TNSListener	0%	4,5 MB
	▷ oravssw	0%	1,8 MB
	▷ RDP Clipboard Monitor	0%	1,6 MB
	▷	0%	5,0 MB



# Oracle Net Listener

---

- ▶ Экземпляр может иметь несколько точек подключения
- ▶ Точки подключения называются сервисами и имеют символические имена
- ▶ При инсталляции автоматически создается два сервиса:
  - ▶ SYS\$USERS (по умолчанию, указывается SID в параметрах соединения),
  - ▶ сервис с именем инстанса (указывается сервис)
- ▶ При создании PDB для нее автоматически добавляется сервис с именем, совпадающим с PDB-именем



# Oracle Net Listener

```
| select name, network_name, pdb from v$services;
```

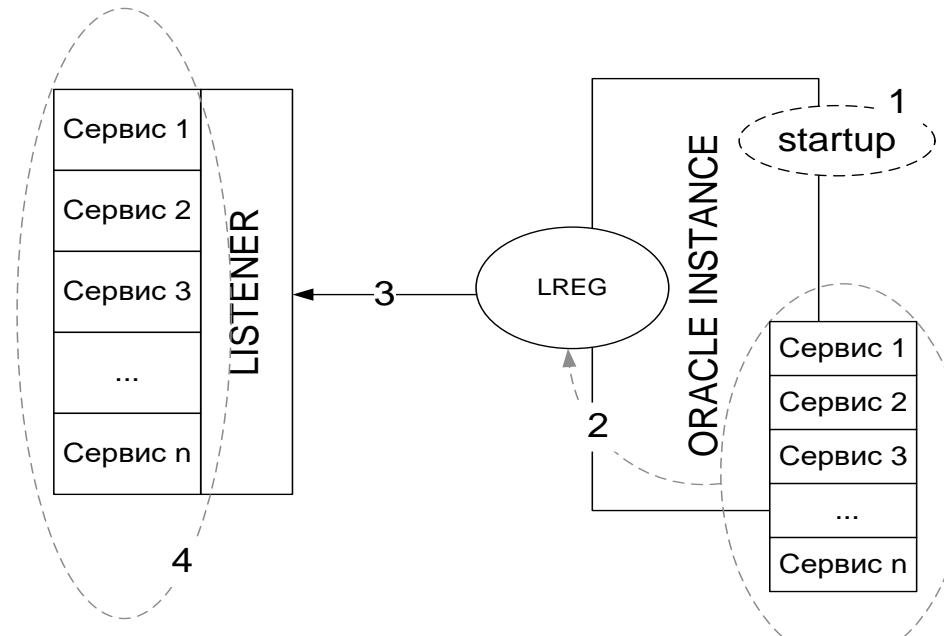
Script Output x Query Result 1 x

SQL | All Rows Fetched: 7 in 0 seconds

NAME	NETWORK_NAME	PDB
1 pdb_b.be.by	pdb_b.be.by	PDB_B
2 pdb_a.be.by	pdb_a.be.by	PDB_A
3 pdborcl.be.by	pdborcl.be.by	PDBORCL
4 orclXDB	orclXDB	CDB\$ROOT
5 orcl.be.by	orcl.be.by	CDB\$ROOT
6 SYS\$BACKGROUND	(null)	CDB\$ROOT
7 SYS\$USERS	(null)	CDB\$ROOT

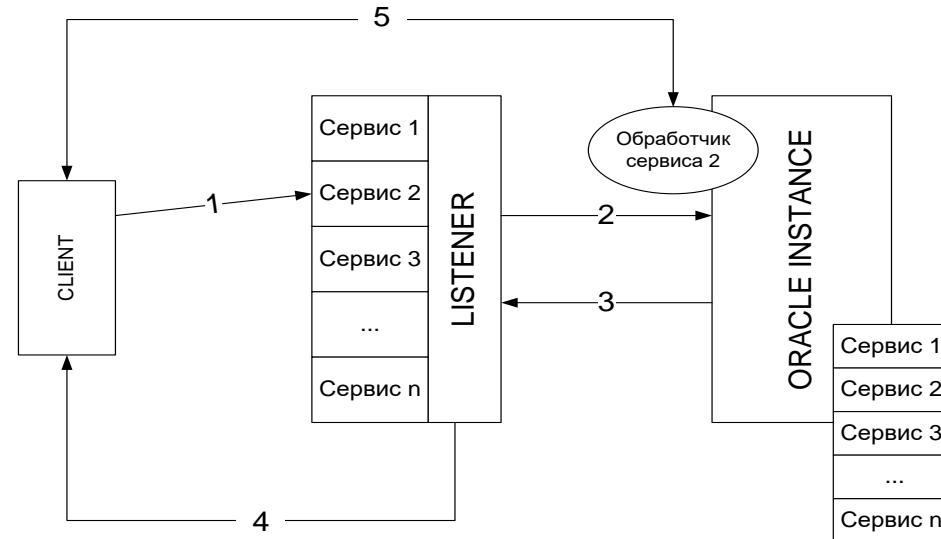
# Oracle Net Listener – старт экземпляра

- ▶ фоновый процесс LREG считывает имена и параметры зарегистрированных сервисов экземпляра
- ▶ регистрирует их в Listener
- ▶ если Listener не обнаружен, то попытки регистрации периодично повторяются
- ▶ Listener прослушивает запросы к сервисам



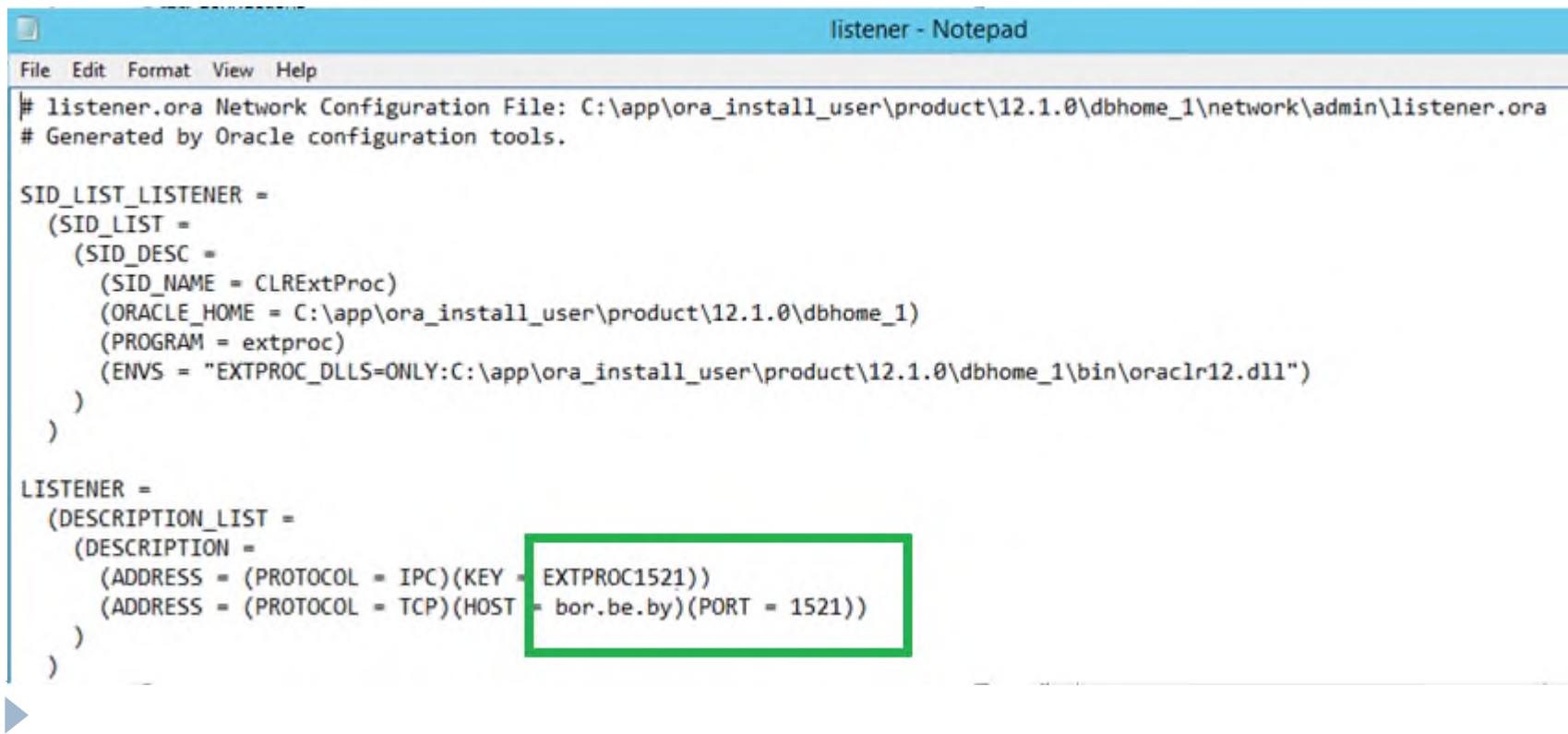
# Oracle Net Listener – запрос на соединение

- ▶ 1) клиент выполняет запрос к Listener на соединение с сервисом экземпляра
- ▶ 2) Listener запрашивает соединение с сервером
- ▶ 3) сервер возвращает параметры соединения с обработчиком сервиса
- ▶ 4) Listener сообщает параметры соединения клиенту
- ▶ 5) клиент соединяется с обработчиком запросов сервиса для дальнейшей работы с сервисом в рамках соединения



# Oracle Net Listener

- ▶ Конфигурационный файл программы Listener:  
**ORACLE\_HOME\NETWORK\ADMIN\listener.ora**
- ▶ Файл считывается при старте Listener



The screenshot shows a Windows Notepad window titled "listener - Notepad". The menu bar includes File, Edit, Format, View, and Help. The content of the file is as follows:

```
# listener.ora Network Configuration File: C:\app\ora_install_user\product\12.1.0\dbhome_1\network\admin\listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\app\ora_install_user\product\12.1.0\dbhome_1)
      (PROGRAM = extproc)
      (ENVS = "EXTPROC_DLLS=ONLY:C:\app\ora_install_user\product\12.1.0\dbhome_1\bin\oraclr12.dll")
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = bor.be.by)(PORT = 1521))
    )
  )
```

A green rectangular box highlights the line "(HOST = bor.be.by)".

# Oracle Net Listener

New / Select Database Connection

Connection Name	Connection Details
eacore_to_pdb_a	eacore@//192.168.1.2...
sys_to_instance_as_s...	sys@//192.168.1.225:...
system_to_instance	system@//192.168.1....

Connection Name: system\_to\_instance  
Username: system  
Password: \*\*\*\*\*

Save Password  Connection Color

**Oracle**

Connection Type: Basic Role: default

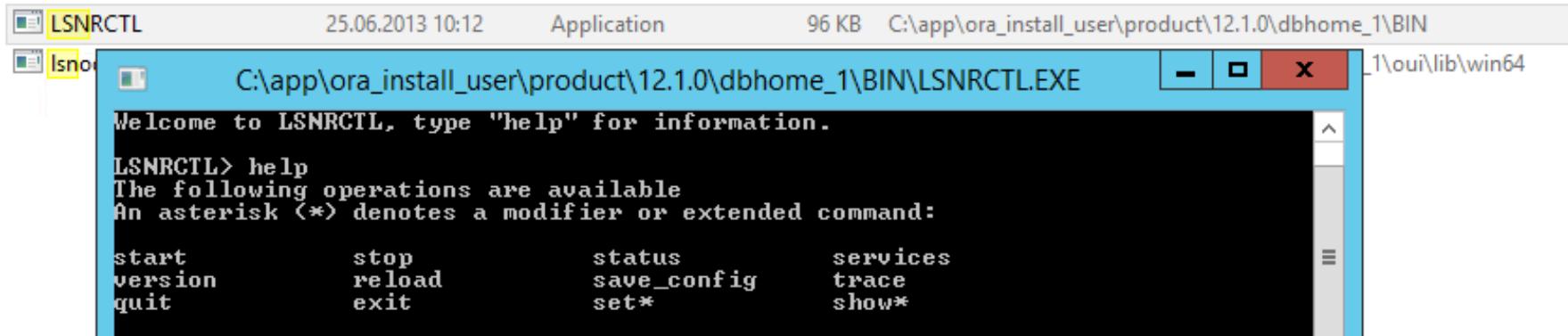
Hostname: bor.be.by  
Port: 1521  
 SID: ord  
 Service name:

OS Authentication  Kerberos Authentication

Status : Success

# Oracle Net Listener

- ▶ Lsnrctl – утилита управления процессом Listener



# Oracle Net Listener

```
C:\app\ora_install_user\product\12.1.0\dbhome_1\BIN\LSNRCTL.EXE
The command completed successfully
LSNRCTL> stop
Connecting to <DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521))>
The command completed successfully
LSNRCTL> _
```

Network Connections				
	Manages Dir...	Running	Manual	Local System...
Network Connectivity Assistant	Provides Dir...		Manual (Trig...	Local Syste...
Network List Service	Identifies th...	Running	Manual	Local Service
Network Location Awareness	Collects an...	Running	Automatic	Network S...
Network Store Interface Service	This service ...	Running	Automatic	Local Service
Optimize drives	Helps the c...		Manual	Local Syste...
OracleJobSchedulerORCL			Disabled	\ora_instal...
OracleOraDB12Home1MTSRecoveryService		Running	Automatic	\ora_instal...
OracleOraDB12Home1TNSListener			Automatic	\ora_instal...
OracleRemExecServiceV2			Manual	Local Syste...
OracleServiceORCL		Running	Automatic	\ora_instal...
OracleVssWriterORCL		Running	Automatic	\ora_instal...
Performance Counter DLL Host	Enables rem...		Manual	Local Service
Performance Logs & Alerts	Performanc...		Manual	Local Service
Plug and Play	Enables a c...	Running	Manual	Local Syste...
Portable Device Enumerator Service	Enforces gr...		Manual (Trig...	Local Syste...
Power	Manages p...	Running	Automatic	Local Syste...
Print Spooler	This service ...	Running	Automatic	Local Syste...
Printer Extensions and Notifications	This service ...		Manual	Local Syste...

# Oracle Net Listener

New / Select Database Connection

Connection Name	Connection Details
eacore_to_pdb_a	eacore@//192.168.1.22...
sys_to_instance_as_sys...	sys@//192.168.1.225:1...
system_to_instance	system@//192.168.1.22...

Connection Name: system\_to\_instance  
Username: system  
Password: \*\*\*\*\*

Save Password  Connection Color

**Oracle**

Connection Type: Basic Role: default

Hostname: bor.be.by  
Port: 1521  
 SID: ord  
 Service name:

OS Authentication  Kerberos Authentication

Status : Failure -Test failed: Ошибка ввода/вывода: The Network Adapter could not establish the connection

Справка     Отмена

# Oracle Net Listener

```
C:\app\ora_install_user\product\12.1.0\dbhome_1\BIN\LSNRCTL.EXE

The command completed successfully
LSNRCTL> stop
Connecting to <DESCRIPTION=<ADDRESS=<PROTOCOL=IPC><KEY=EXTPROC1521>>>
The command completed successfully
LSNRCTL> start
Starting tnslsnr: please wait...

TNSLSNR for 64-bit Windows: Version 12.1.0.1.0 - Production
System parameter file is C:\app\ora_install_user\product\12.1.0\dbhome_1\network
\admin\listener.ora
Log messages written to C:\app\ora_install_user\diag\tns\lsnr\b0r\listener\alert\
log.xml
Listening on: <DESCRIPTION=<ADDRESS=<PROTOCOL=ipc><PIPENAME=\.\pipe\EXTPROC1521
ipc>>>
Listening on: <DESCRIPTION=<ADDRESS=<PROTOCOL=tcp><HOST=b0r.be.by><PORT=1521>>>

Connecting to <DESCRIPTION=<ADDRESS=<PROTOCOL=IPC><KEY=EXTPROC1521>>>
STATUS of the LISTENER
-----
Alias                      LISTENER
Version                    TNSLSNR for 64-bit Windows: Version 12.1.0.1.0 - Produ
ction
Start Date                 11-OCT-2016 09:52:51
Uptime                     0 days 0 hr. 0 min. 6 sec
Trace Level                off
Security                   ON: Local OS Authentication
SNMP                       OFF
Listener Parameter File   C:\app\ora_install_user\product\12.1.0\dbhome_1\networ
k\admin\listener.ora
Listener Log File          C:\app\ora_install_user\diag\tns\lsnr\b0r\listener\aler
t\log.xml
Listening Endpoints Summary...
  <DESCRIPTION=<ADDRESS=<PROTOCOL=ipc><PIPENAME=\.\pipe\EXTPROC1521 ipc>>>
  <DESCRIPTION=<ADDRESS=<PROTOCOL=tcp><HOST=b0r.be.by><PORT=1521>>>
Services Summary...
Service "CLRExtProc" has 1 instance(s).
  Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

# Oracle Net Listener

OracleJobSchedulerORCL	Disabled	\ora_instal...
OracleOraDB12Home1MTSRecoveryService	Running	Automatic
OracleOraDB12Home1TNSListener	Running	Automatic
OracleRemExecServiceV2	Manual	Local Syste...
OracleServiceORCL	Running	Automatic
OracleVssWriterORCL	Running	Automatic

New / Select Database Connection

X

Connection Name	Connection Details
eacore_to_pdb_a	eacore@//192.168.1.2...
sys_to_instance_as_s...	sys@//192.168.1.225:...
system_to_instance	system@//bor.bte.by:...

Connection Name: system\_to\_instance  
Username: system  
Password:   Save Password  Connection Color

**Oracle**

Connection Type: Basic Role: default

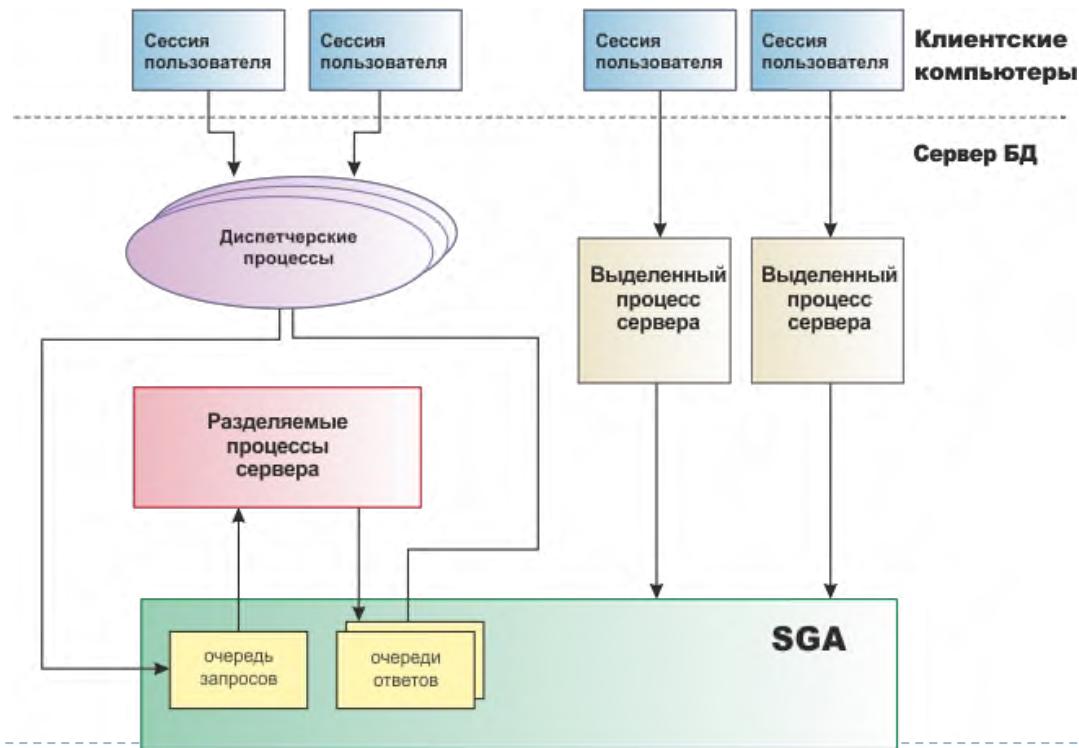
Hostname: bor.be.by  
Port: 1521  
 SID: ord  
 Service name:

OS Authentication  Kerberos Authentication

Status : Success

# Серверные процессы

- ▶ Серверные процессы выполняют работу от имени сеанса клиента
  - ▶ Подключение посредством выделенного сервера
  - ▶ Подключение при помощи разделяемого сервера



# Режимы сессий

```
.....  
Select username, sid, serial#, server, paddr, status  from v$session where username is not null;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 3 in 0 seconds

	USERNAME	SID	SERIAL#	SERVER	PADDR	STATUS
1	SYSTEM	43	807	DEDICATED	00007FFB8AA43C98	ACTIVE
2	A2	54	11753	DEDICATED	00007FFB8AA42B08	INACTIVE
3	SYS	78	7849	DEDICATED	00007FFB8AA44E28	INACTIVE

# Dedicated server

```
select paddr, username, service_name, server, osuser, machine, program from v$session where username is not null;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 3 in 0 seconds

PADDR	USERNAME	SERVICE_NAME	SERVER	OSUSER	MACHINE	PROGRAM
1 00007FFB8AA43C98	SYSTEM	SYS\$USERS	DEDICATED	ea	bor	SQL Developer
2 00007FFB8AA42B08	A2	pdb_a.be.by	DEDICATED	ea	bor	SQL Developer
3 00007FFB8AA44E28	SYS	SYS\$USERS	DEDICATED	BTE\ea	BE\BOR	sqlplus.exe

```
select addr, spid, pname from v$process where background is null order by pname;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 10 in 0,016 seconds

ADDR	SPID	PNAME
1 00007FFB8AA30078	1136	D000
2 00007FFB8AA39E88	1648	P000
3 00007FFB8AA3B018	2760	P001
4 00007FFB8AA3C1A8	2456	P002
5 00007FFB8AA3D338	1740	P003
6 00007FFB8AA31208	2332	S000
7 00007FFB8AA1D5E8	(null)	(null)
8 00007FFB8AA44E28	1852	(null)
9 00007FFB8AA42B08	788	(null)
10 00007FFB8AA43C98	1916	(null)

# Shared server

---

- ▶ Уменьшается количество серверных процессов
- ▶ Предпочтительнее, когда в системе присутствует большое количество пользователей
- ▶ Для длительных заданий лучше использовать выделенный режим – почему?



# Параметры dispatcher

```
show parameter dispatcher
```

NAME	TYPE	VALUE
------	------	-------

dispatchers	string	(protocol=tcp)(dispatchers=3)(service=isus_a,isusxdb)
max_dispatchers	integer	

```
alter system set max_dispatchers=10;
```

```
show parameter dispatcher;
```

dispatchers	string	(protocol=tcp)(dispatchers=3)(service=isus_a,isusxdb)
max_dispatchers	integer	10



Активаци



# Параметры dispatcher

## MAX\_DISPATCHERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If MAX_DISPATCHERS is specified, then it should be greater than or equal to the number of dispatchers specified by the DISPATCHERS parameter and less than the number of processes specified by the PROCESSES parameter.
Basic	No

## MAX\_SHARED\_SERVERS

Property	Description
Parameter type	Integer
Default value	There is no default value.
Modifiable	ALTER SYSTEM
Range of values	If MAX_SHARED_SERVERS is specified, then it should be greater than or equal to SHARED_SERVERS and less than PROCESSES.
Basic	No



# Параметры dispatcher

```
alter system set shared_servers=5;
select name, status, messages, idle, busy from v$shared_server;
```

NAME	STATUS	MESSAGES	IDLE	BUSY
S000	WAIT (COMMON)	238	5508300	9561
S001	WAIT (COMMON)	0	13359	1
S002	WAIT (COMMON)	0	13358	0
S003	WAIT (COMMON)	0	13357	0
S004	WAIT (COMMON)	0	13356	0

```
select pname, program from v$process where background is null order by pname
```

PNAME	PROGRAM
D000	ORACLE.EXE (D000)
D001	ORACLE.EXE (D001)
D002	ORACLE.EXE (D002)
P000	ORACLE.EXE (P000)
P001	ORACLE.EXE (P001)
P002	ORACLE.EXE (P002)
P003	ORACLE.EXE (P003)
S000	ORACLE.EXE (S000)
S001	ORACLE.EXE (S001)
S002	ORACLE.EXE (S002)
S003	ORACLE.EXE (S003)
S004	ORACLE.EXE (S004)
(null)	PSEUDO
(null)	ORACLE.EXE (SHAD)
(null)	ORACLE.EXE (SHAD)
(null)	ORACLE.EXE (SHAD)

# Сравнение режимов выделенного и разделяемого серверов

---

- ▶ Настройка
- ▶ Установление подключений
- ▶ Режим работы
- ▶ Короткие и длинные транзакции
- ▶ Запуск и останов БД – только в режиме выделенного сервера



# Фоновые процессы

- ▶ Фоновые (background) процессы – специальная группа процессов для обеспечения производительности и поддержки работы большого числа пользователей

LREG	DBWn
CKRT	LGWR
PMON	SMON
RECO	ARCn



# Фоновые процессы

```
select count(*) from v$bgprocess;

select name, description from v$bgprocess where paddr!=hextoraw('00') order by name;
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x

SQL | All Rows Fetched: 1 in 0 seconds

	COUNT(*)	NAME	DESCRIPTION
1	401	AQPC	AQ Process Coord
2	CJQ0	Job Queue Coordinator	
3	CKPT	checkpoint	
4	DBRM	DataBase Resource Manager	
5	DBW0	db writer process 0	
6	DIAG	diagnosibility process	
7	DIA0	diagnosibility process 0	
8	GEN0	generic0	
9	LGWR	Redo etc.	
10	LREG	Listener Registration	
11	MMAN	Memory Manager	
12	MMNL	Manageability Monitor Process 2	
13	MMON	Manageability Monitor Process	
14	PMON	process cleanup	
15	PSP0	process spawner 0	
16	RECO	distributed recovery	

# LREG

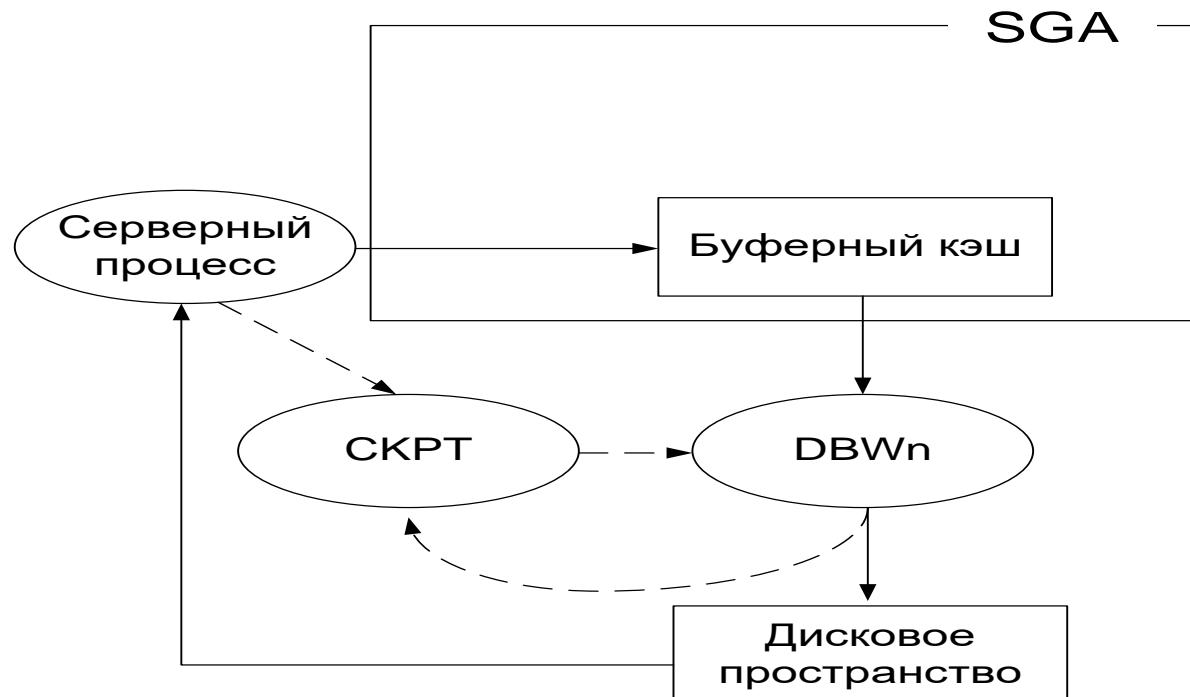
---

- ▶ Listener Registration Process – периодическая регистрация сервисов в процессе Listener



# DBWn

- ▶ Database Writer Process: ( $n=0,\dots,9, a,\dots,z; BWm, m=36,\dots,99$ ) – фоновый процесс записывающий по LRU измененные блоки (грязные блоки) в файлы базы данных.



## DBWn

---

- ▶ В большинстве случаев достаточно одного процесса
- ▶ Если процессор 1, то в более чем в 1 DBW не смысла
- ▶ Проверяет с периодичностью не менее 3 сек. наличие измененных блоков и инициирует *checkpoint* и по команде фонового процесса **СКРТ** сбрасывает некоторую их часть на диск
- ▶ Использует асинхронный ввод-вывод для записи блоков на диск



## DBWn

---

- ▶ Сброс измененных блоков буферного кэша может инициироваться фоновым процессом **СКРТ**
- ▶ Сброс измененных блоков может инициироваться через СКРТ серверным процессом (выделенным или разделяемым), который не обнаружил свободного места в буферном кэше
- ▶ DBW записывает измененные (LRU, RECYCLE/DEFAULT/KEEP) блоки на диск и освобождает место в буферном кэше



# СКРТ

---

- ▶ СКРТ выполняет процесс `checkpoint`
- ▶ Выполняется при `shutdown`, `alter system checkpoint`, переключении REDO – журнала, периодическом сообщение от DWR, backup
- ▶ записывает информацию о контрольной точке в управляющие файлы
- ▶ дает команду DBW на сброс буферов
- ▶ дает команду LGWR на сброс буферов



# СКРТ

---

- ▶ инициирует создание контрольной точки
- ▶ изменяет управляющие файлы (последний SCN)
- ▶ изменяет заголовки файлов
- ▶ сигнализирует DBWR о необходимости сбросить буферы

```
alter system checkpoint; -- создать контрольную точку
```



# LGWR

---

- ▶ Log Writer Process – только один
- ▶ управляет буфером журналов повтора
- ▶ записывает блоки буфера журналов повтора в группы журналов
- ▶ записывает изменения базы данных до их фиксации DBWR в базе данных
- ▶ переключает текущую группу
  - ▶ Раз в 3 секунды
  - ▶ При фиксации транзакции
  - ▶ При заполнении буфера на 1/3 или записи в него 1Мб



# LGWR

---

- ▶ Если по какой-то причине LGWR не может записать данные в один из файлов группы, он продолжает работать (информация об этом событии в alert)
- ▶ Если файл в группе один или нет возможности записать ни в один файл группы, то экземпляр остановится
- ▶ Производительность LGWR является критической для экземпляра



# LGWR

---

- ▶ Инициирует создание контрольных точек
- ▶ Чем чаще контрольные точки, тем меньше времени потребуется при восстановлении базы данных при сбое



# Параметры LGWR

## LOG\_CHECKPOINT\_INTERVAL

Property	Description
Parameter type	Integer
Default value	0
Modifiable	ALTER SYSTEM
Range of values	0 to $2^{31} - 1$
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LOG_CHECKPOINT_INTERVAL` specifies the frequency of checkpoints in terms of the number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log. This number refers to physical operating system blocks, not database blocks.



# Параметры LGWR

## LOG\_CHECKPOINT\_TIMEOUT

Property	Description
Parameter type	Integer
Default value	1800
Modifiable	ALTER SYSTEM
Range of values	0 to $2^{31} - 1$
Basic	No
Real Application Clusters	Multiple instances can have different values.

`LOG_CHECKPOINT_TIMEOUT` specifies (in seconds) the amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log (sometimes called the **tail of the log**) occurred. This parameter also signifies that no buffer will remain dirty (in the cache) for more than *integer* seconds.



# Параметры LGWR

```
| show parameter log_checkpoint
```

NAME	TYPE	VALUE
log_checkpoint_interval	integer	0
log_checkpoints_to_alert	boolean	FALSE
log_checkpoint_timeout	integer	1800



# ARCn

---

- ▶ Archiver Process – копирует файлы журнала повтора после переключения группы журналов
- ▶ Необязательный процесс



# ARCn

---

## LOG\_ARCHIVE\_DEST

---

Property	Description
Parameter type	String
Syntax	<code>LOG_ARCHIVE_DEST = filespec</code>
Default value	Null
Modifiable	<code>ALTER SYSTEM</code>
Range of values	Any valid path or device name, except raw partitions
Basic	No



# ARCn

---

## LOG\_ARCHIVE\_FORMAT

Property	Description
Parameter type	String
Syntax	<code>LOG_ARCHIVE_FORMAT = filename</code>
Default value	Operating system-dependent
Modifiable	No
Range of values	Any string that resolves to a valid filename

`%s` log sequence number

`%s` log sequence number, zero filled

`%t` thread number

`%T` thread number, zero filled

`LOG_ARCHIVE_FORMAT = 'log%t_%s_%r.arc'`

`%a` activation ID

`%d` database ID

`%r` resetlogs ID that ensures unique names are constructed for the archived log files across  
of the database



# PMON

---

- ▶ Process monitor – отвечает за очистку после ненормального закрытия подключений
- ▶ Инициирует откат незафиксированных транзакций, снятие блокировок, и освобождение ресурсов SGA
- ▶ следит за работой других фоновых процессов, отвечает за их перезапуск
- ▶ восстанавливает работу dispatcher или shadow процессов при системном сбое



# SMON

---

- ▶ System Monitor Process – системный монитор
- ▶ Восстановление экземпляра для узла
- ▶ Восстановление незавершенных транзакций
- ▶ Очистка временных сегментов данных
- ▶ Очистка временных табличных пространств
- ▶ Объединение свободного пространства
- ▶ Очистка таблицы OBJ
- ▶ Сжатие сегментов отката



# RECO

---

- ▶ Recovery Process – разрешение проблем связанных с распределенными транзакциями



# RECO

---

- ▶ Одна из баз данных, к которой первоначально подключился клиент, становится координатором
- ▶ Сервер опрашивает остальные  $N - 1$  серверов, готовы ли они фиксировать транзакцию
- ▶ Каждый из  $N - 1$  серверов сообщает о своем состоянии готовности как да (**YES**) или нет (**NO**)
- ▶ Если любой из серверов вернул **NO**, вся транзакция откатывается
- ▶ Если все серверы вернули **YES**, координатор рассыпает всем  $N - 1$  серверам сообщение о постоянной фиксации



# RECO

---

- ▶ Если серверы ответили **YES** и происходит сбой сети транзакция становится *сомнительной* (*in-doubt*) распределенной транзакцией
- ▶ Протокол обработки распределенных транзакций не может полностью предотвратить сомнительные транзакции
- ▶ Остальную обработку сомнительной транзакции выполняет процесс **RECO**



# RECO

---

- ▶ Транзакция остается незафиксированной
- ▶ **RECO** может восстановить либо откатить транзакцию, связавшись с координатором
- ▶ Если процесс **RECO** не может связаться с координатором, то необходимо зафиксировать или откатить транзакции вручную



# FBDA

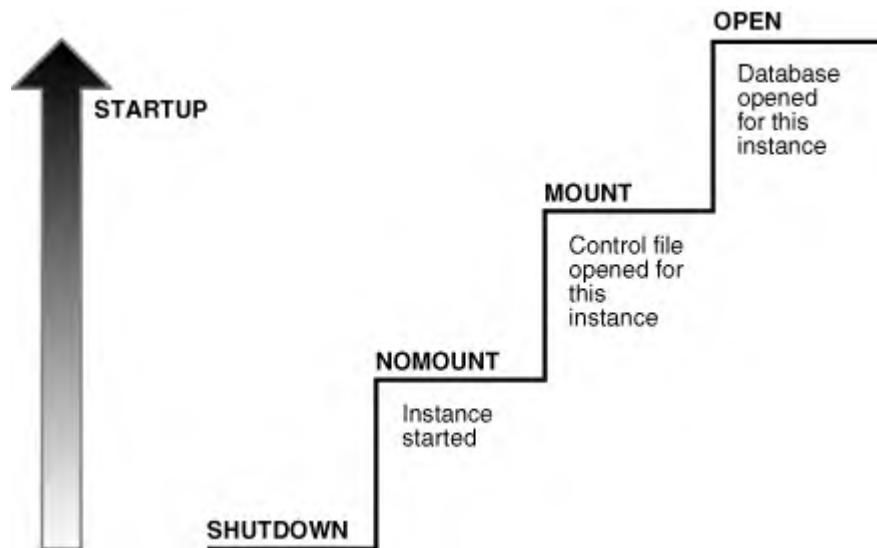
---

- ▶ Flashback Data Archiever – архивирование ретроспективных данных



# Режимы старта и останова экземпляра

- ▶ Для запуска или остановки экземпляра должно использоваться подключение с разрешением SYSDBA или SYSOPER



# Старт экземпляра базы данных Oracle

---

- ▶ STARTUP NOMOUNT
- ▶ STARTUP MOUNT
- ▶ STARTUP OPEN
- ▶ STARTUP FORCE
- ▶ STARTUP RESTRICT



# Останов экземпляра базы данных

---

- ▶ SHUTDOWN NORMAL
- ▶ SHUTDOWN TRANSACTIONAL
- ▶ SHUTDOWN IMMEDIATE
- ▶ SHUTDOWN ABORT



# STARTUP NOMOUNT

---

- ▶ Запуск экземпляра Oracle без монтирования базы данных
- ▶ Доступ пользователей запрещен
- ▶ Используется для создания БД и пересоздания управляющих файлов



# STARTUP MOUNT

---

- ▶ Запустить экземпляр, монтировать БД, но не запускать ее
- ▶ Считан файл параметров
- ▶ Происходит обращение к управляющим файлам
- ▶ Проверяется состояние файлов БД
- ▶ Доступ пользователей запрещен
- ▶ Из состояния **STARTUP NOMOUNT** переводится **ALTER DATABASE MOUNT**



# STARTUP OPEN

---

- ▶ Запустить экземпляр
- ▶ Монтировать и открыть БД
- ▶ Могут присоединяться пользователи
- ▶ Из состояния STARTUP MOUNT переводится ALTER DATABASE OPEN
- ▶ Можно стартовать в режиме только для чтения
  - ▶ ALTER DATABASE OPEN READ ONLY
  - ▶ ALTER DATABASE OPEN READ WRITE
- ▶ STARTUP RESTRICT - для пользователей, имеющих привилегию RESTRICTED SESSION
- ▶ ALTER SYSTEM DISABLE(ENABLE) RESTRICTED SESSION



# SHUTDOWN NORMAL

---

- ▶ Запрещено создавать новые сессии
- ▶ Ожидается завершение работы всех пользователей
- ▶ Самый безопасный и долгий способ останова
- ▶ Никаких восстановительных работ при следующем старте не проводится



# SHUTDOWN TRANSACTIONAL

---

- ▶ Запрещено создавать новые сессии
- ▶ Запрещено запускать новые транзакции
- ▶ Сервер дожидается завершения уже начатых транзакций и отключает пользователей, не имеющих активных транзакций
- ▶ Применяется в случаях, когда нет возможности применить **NORMAL**
- ▶ Никаких восстановительных работ при следующем старте не проводится



## SHUTDOWN IMMEDIATE

---

- ▶ Запрещено создавать новые сессии
- ▶ Запрещено запускать новые транзакции
- ▶ Все незафиксированные транзакции откатываются
- ▶ Применяется в случаях, когда нет возможности ждать
- ▶ Никаких восстановительных работ при следующем старте не проводится



# SHUTDOWN ABORT

---

- ▶ Применяется в крайних случаях, когда остальные режимы останова не приводят к результату
- ▶ Все действия прекращаются
- ▶ Все транзакции не фиксируются и не откатываются
- ▶ Пользователей отсоединяют от БД
- ▶ При следующем старте будет выполнено возможное восстановление



# Вопросы?

---



# **Администрирование баз данных и приложений**

**Установление соединения по сети**

**Лекция 7**

# Типы соединений клиента с сервером

- ▶ Oracle Net Services – набор служб, которые устанавливают подключение между сервером БД и пользователями БД
  - ▶ Службы Oracle Net
  - ▶ Oracle Net Listener
  - ▶ Oracle Net Configuration Assistant
  - ▶ Oracle Net Manager
  - ▶ Oracle Connection Manager



Net Configuration Assistant



Net Manager

# Oracle Net

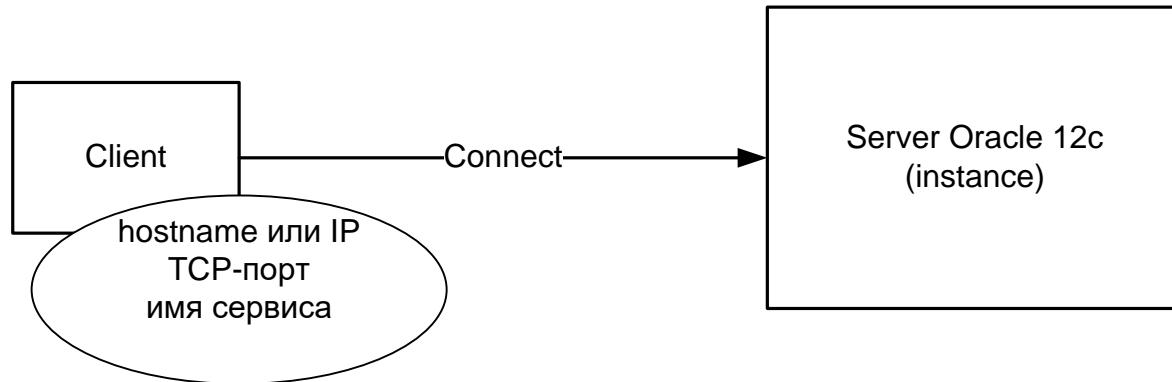
---

- ▶ Oracle Net – программный компонент, который инициализирует, устанавливает и поддерживает подключения между клиентом и сервером.
- ▶ Должен быть установлен и на клиенте, и на сервере.
- ▶ Состоит из двух компонентов:
  - ▶ Oracle Network Foundation layer – отвечает за установку и поддержание подключений между клиентским приложением и сервером.
  - ▶ Oracle Protocol Support – отвечает за отображение функциональности TNS (Transparent Network Substrate) на стандартные протоколы, используемые при подключении.



# Соединение клиента с сервером

- ▶ Имена экземпляров (SID)
- ▶ Имена служб – глобальные имена баз данных
- ▶ Дескрипторы соединений
- ▶ Идентификаторы соединений
- ▶ Строки соединений



# Дескрипторы соединений

---

- ▶ Дескриптор соединения – объединенная спецификация двух обязательных компонентов подключения к базе данных:
  - ▶ Имени службы базы данных
  - ▶ Местоположения адреса базы данных

```
(DESCRIPTION
  (ADDRESS = (PROTOCOL = TCP)
    (HOST = имя_хоста)
    (PORT = 1521))
  (CONNECT_DATA =
    (SERVICE_NAME = имя_службы_базы_данных)))
```

```
(DESCRIPTION
  (ADDRESS = (PROTOCOL = TCP)
    (HOST = 192.168.1.225)(PORT = 1521))
  (CONNECT_DATA = (SERVICE_NAME = pdb_a.be.by)))
```



# Идентификаторы соединений

---

```
 pdb_a = (DESCRIPTION
    (ADDRESS = (PROTOCOL = TCP)
        (HOST = 192.168.1.225)(PORT = 1521))
    (CONNECT_DATA = (SERVICE_NAME = pdb_a.be.by)))
```



# Строки соединений

---

- ▶ Подключение к базе данных выполняется путем указания строки соединения:

```
CONNECT scott/tiger@(DESCRIPTION  
  (ADDRESS = (PROTOCOL = TCP)  
    (HOST = 192.168.1.225)(PORT = 1521))  
  (CONNECT_DATA = (SERVICE_NAME = pdb_a.be.by)))
```

```
CONNECT scott/tiger@pdb_a
```



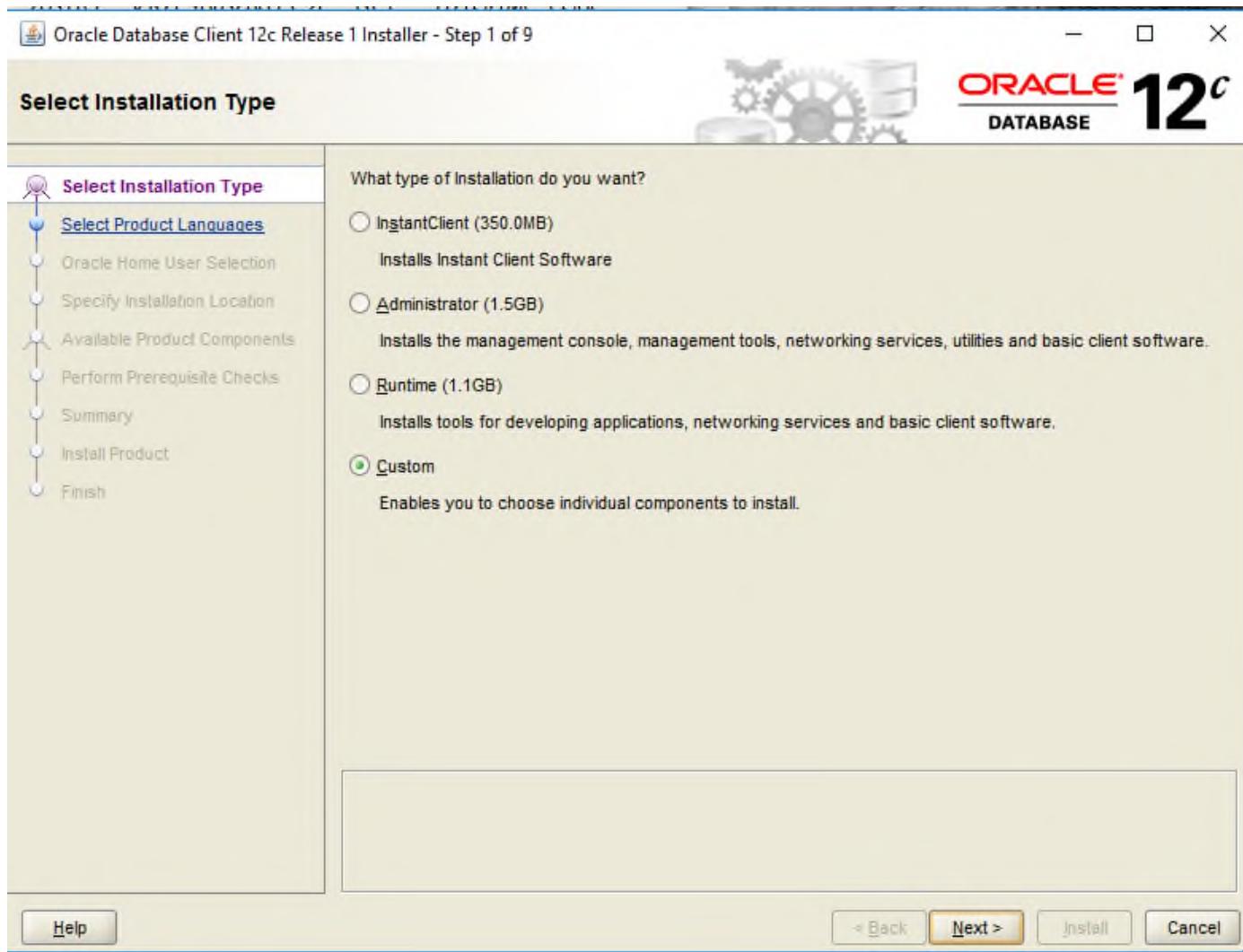
# Oracle - клиент

---

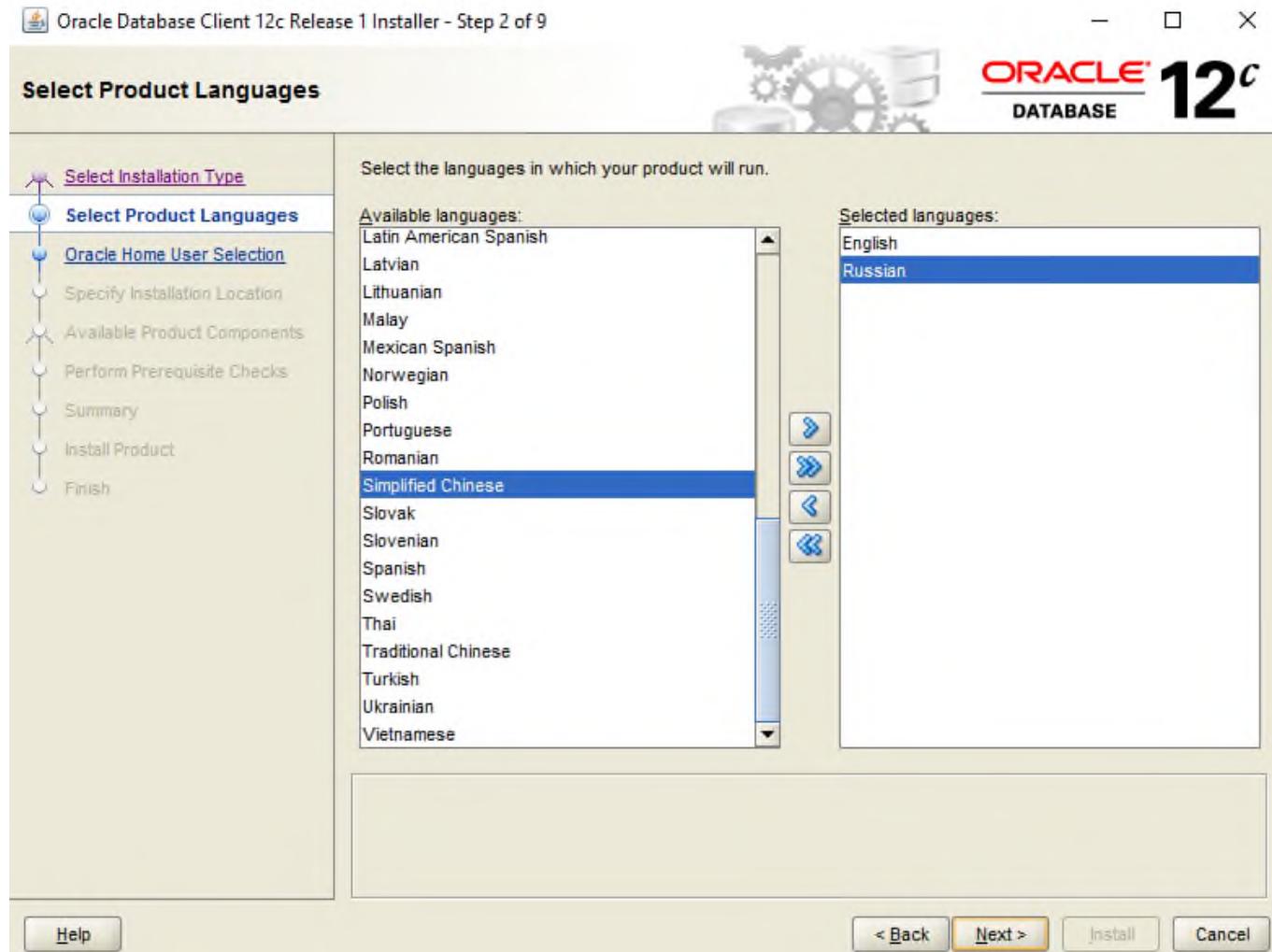
- ▶ Есть в поставке сервера
- ▶ Дополнительно загружается с oracle.com
- ▶ Не обязательно должен совпадать по версии с сервером, но желательно



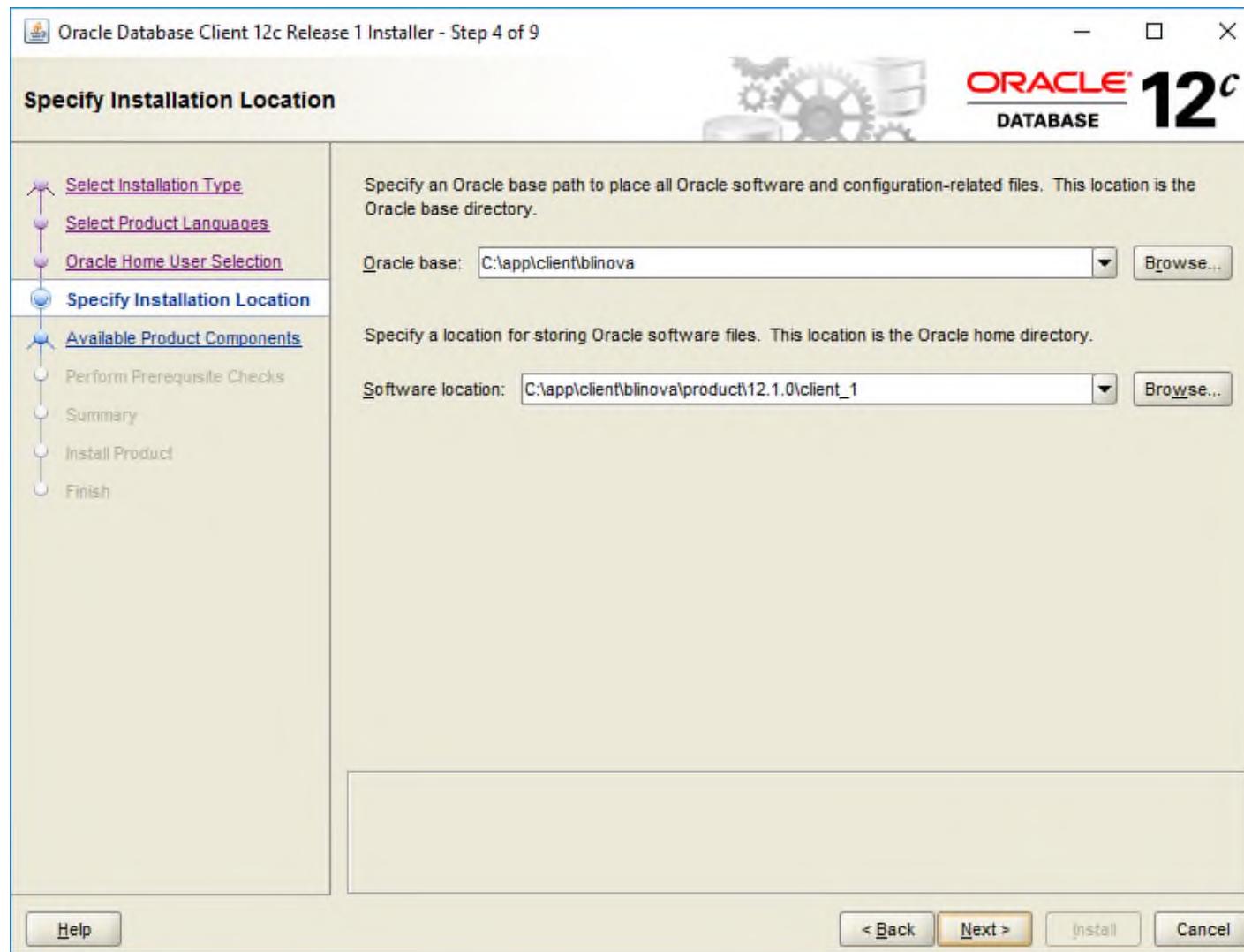
# Oracle - клиент



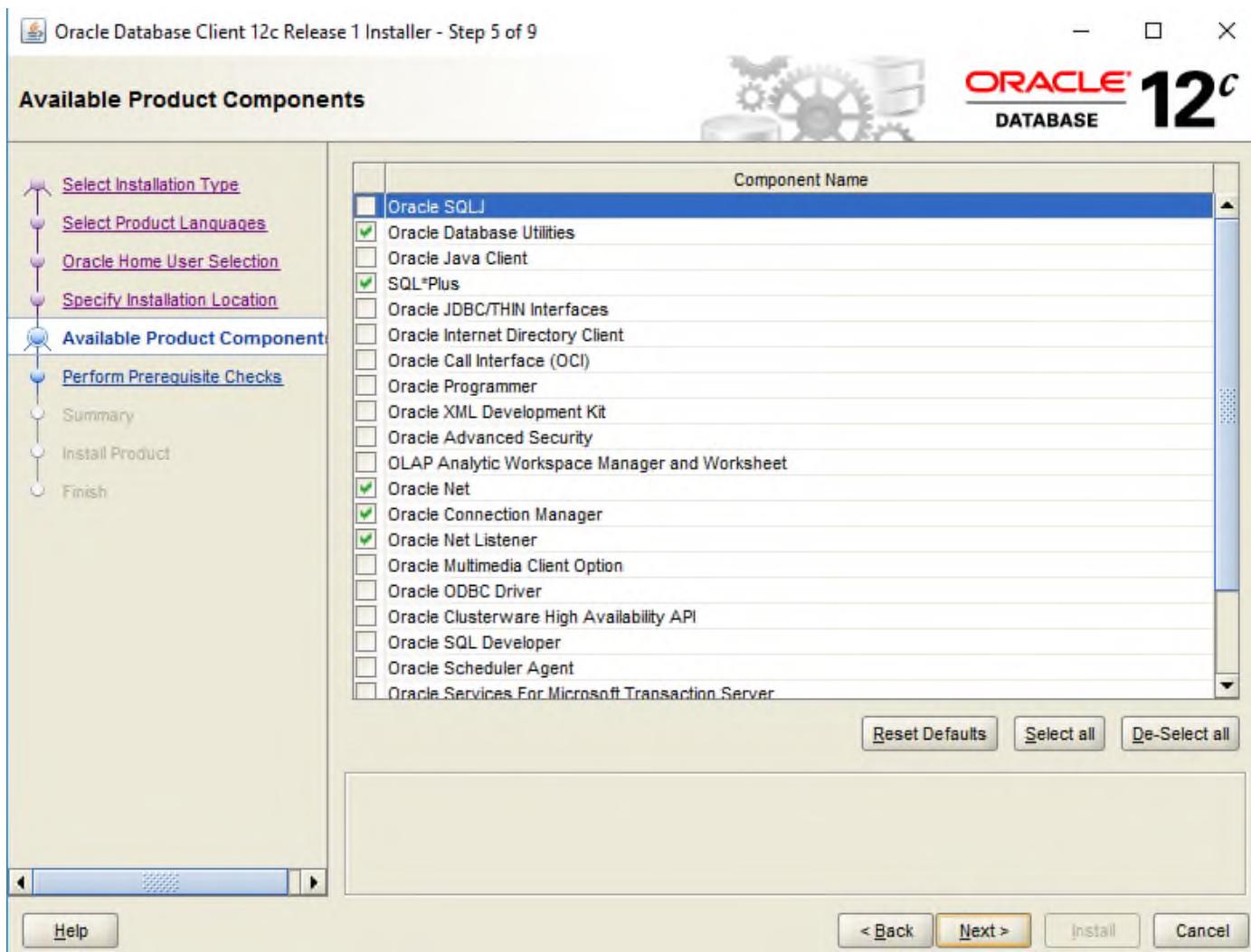
# Oracle - клиент



# Oracle - клиент

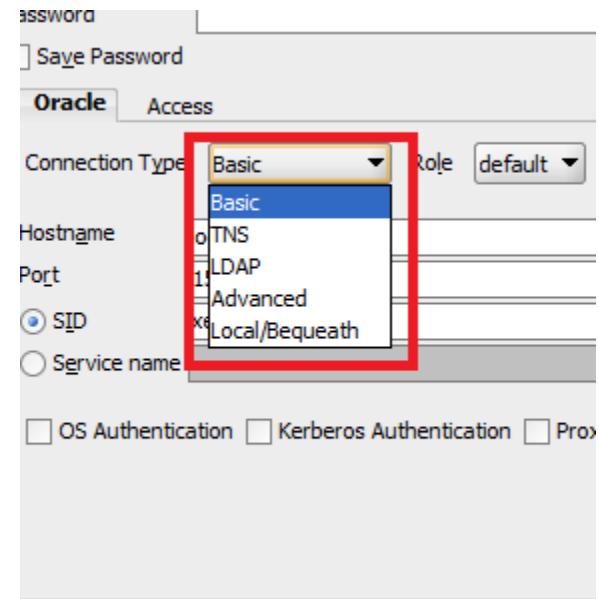


# Oracle - клиент



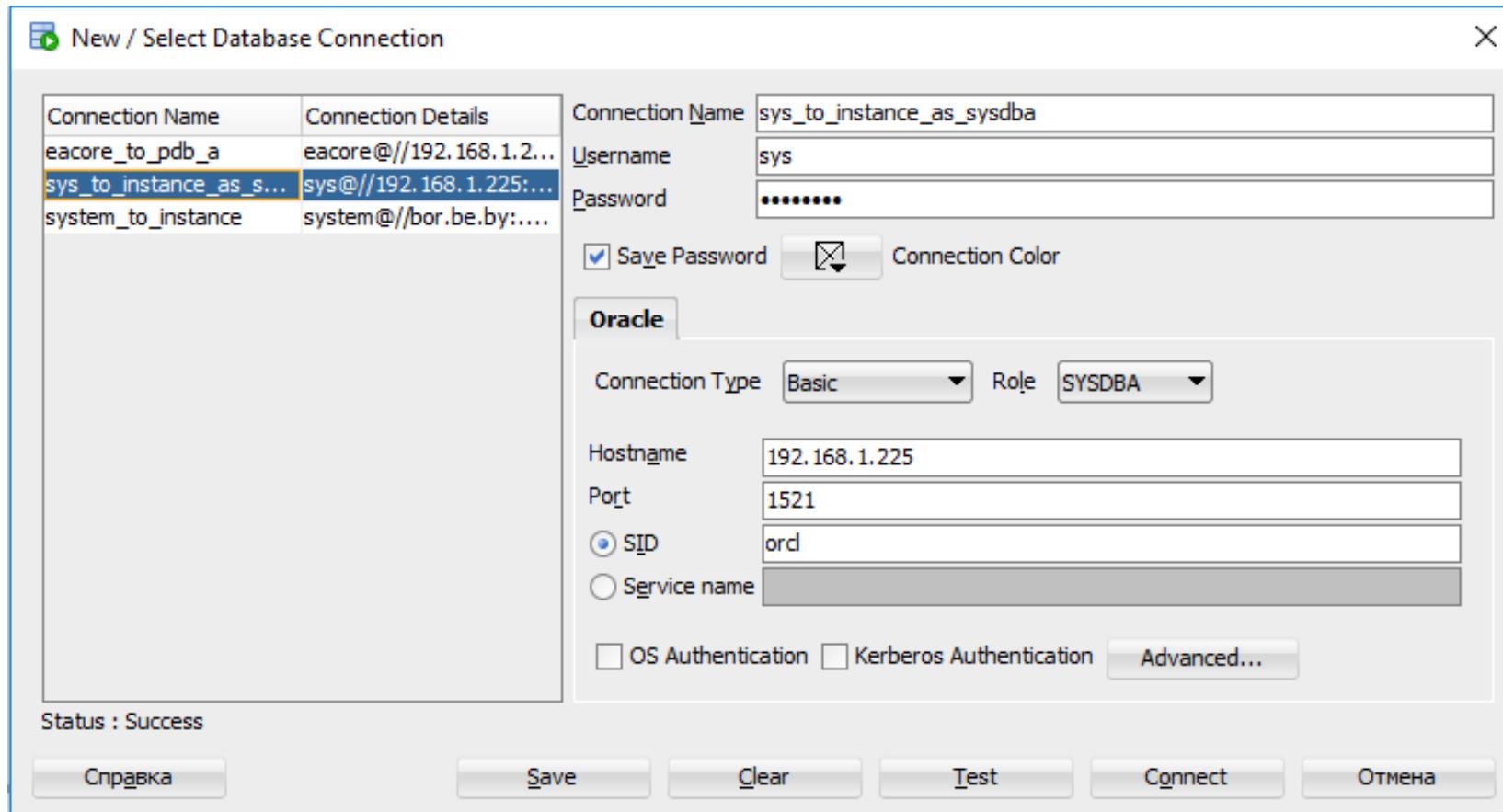
# Виды подключений к Oracle

- ▶ Простое подключение – Basic
- ▶ Локальное именование – TNS
- ▶ LDAP-соединение
- ▶ Local/bequeath-соединение
- ▶ Прочие



# Basic – соединение

- ▶ Явно указываются все параметры соединения



# Basic – соединение

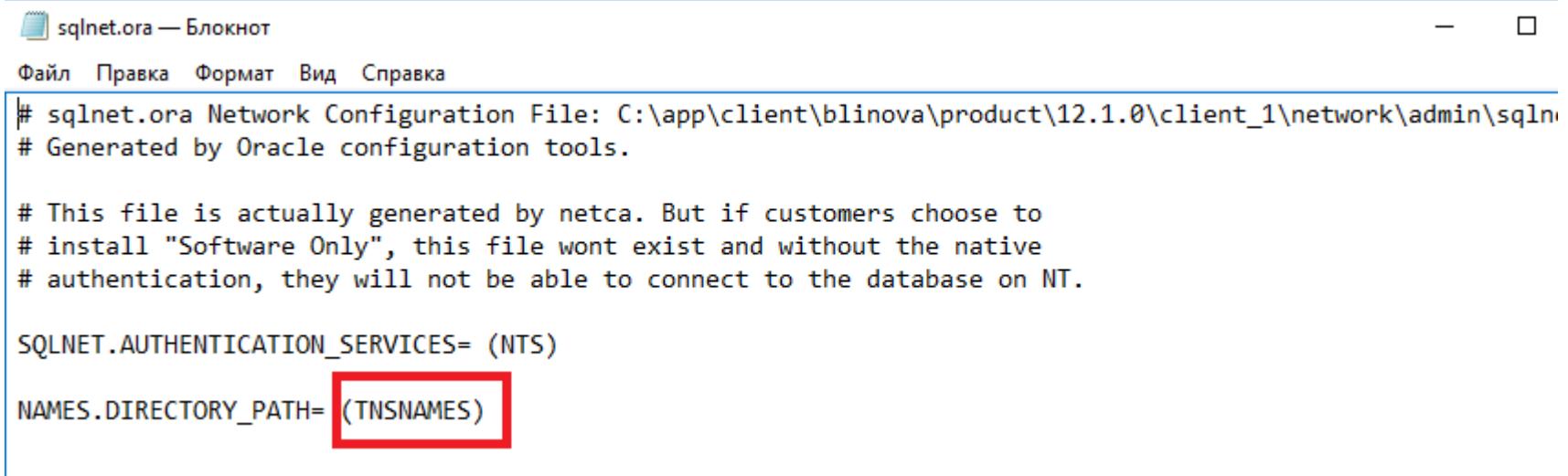
```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 25 17:23:58 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect system/Pa$$w0rd@192.168.1.225:1521/pdb_a.be.by
ERROR:
ORA-12154: TNS:could not resolve the connect identifier specified

SQL>
```

```
SQL> connect system/Pa$$w0rd@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.1.225)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=pdb_a.be.by)))
Connected.
SQL>
```

# Basic – соединение

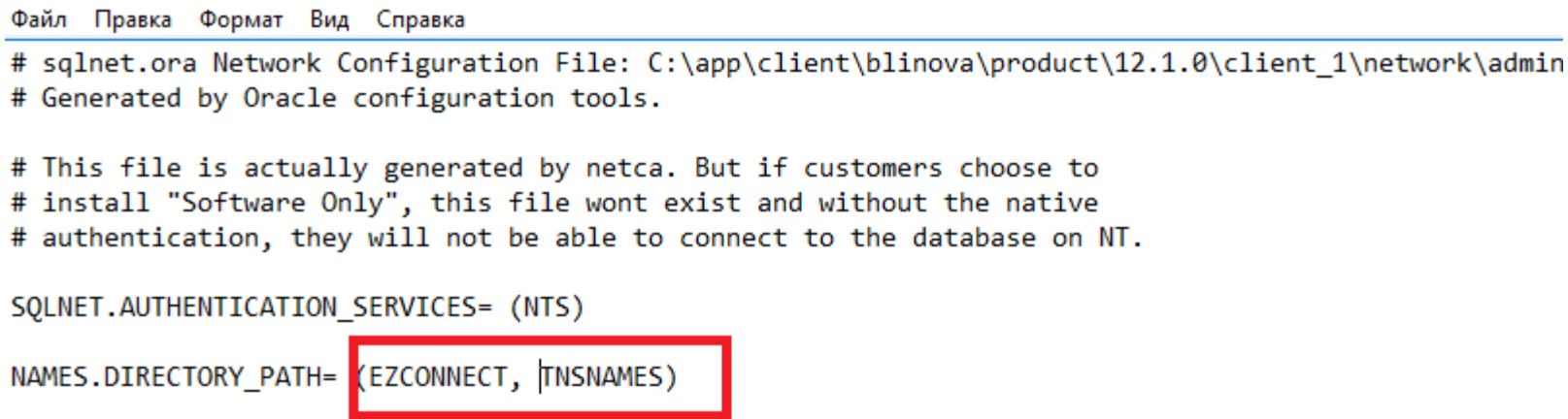


```
sqlnet.ora — Блокнот
Файл Правка Формат Вид Справка
# sqlnet.ora Network Configuration File: C:\app\client\blinova\product\12.1.0\client_1\network\admin\sqln...
# Generated by Oracle configuration tools.

# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)

NAMES.DIRECTORY_PATH= (TNSNAMES)
```



```
Файл Правка Формат Вид Справка
# sqlnet.ora Network Configuration File: C:\app\client\blinova\product\12.1.0\client_1\network\admin
# Generated by Oracle configuration tools.

# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)

NAMES.DIRECTORY_PATH= (EZCONNECT, TNSNAMES)
```



# Basic – соединение

- ▶ **CONNECT имя/пароль@[//]хост[:порт][/имя\_службы]**

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 25 16:08:35 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect system/Pa$$w0rd@192.168.1.225:1521/pdb_a.be.by
Connected.
SQL> ■
```

```
SQL> connect system/Pa$$w0rd@192.168.1.225/pdb_a.be.by
Connected.
SQL>
```

## Basic – соединение

---

- ▶ Должны быть установлены Oracle Net Services
- ▶ Поддержка протокола TCP/IP – на сервере и клиенте
- ▶ Нельзя использовать расширенные сетевые функциональные возможности Oracle



# TNS-соединение

ent > blinova > product > 12.1.0 > client\_1 > NETWORK > ADMIN

Имя	Дата изменения
SAMPLE	25.10.2016 13:58
listener.ora	25.10.2016 14:00
sqlnet.ora	25.10.2016 17:34
sqlnet1610256AM2941.bak	25.10.2016 13:59
tnsnames.ora	25.10.2016 14:01

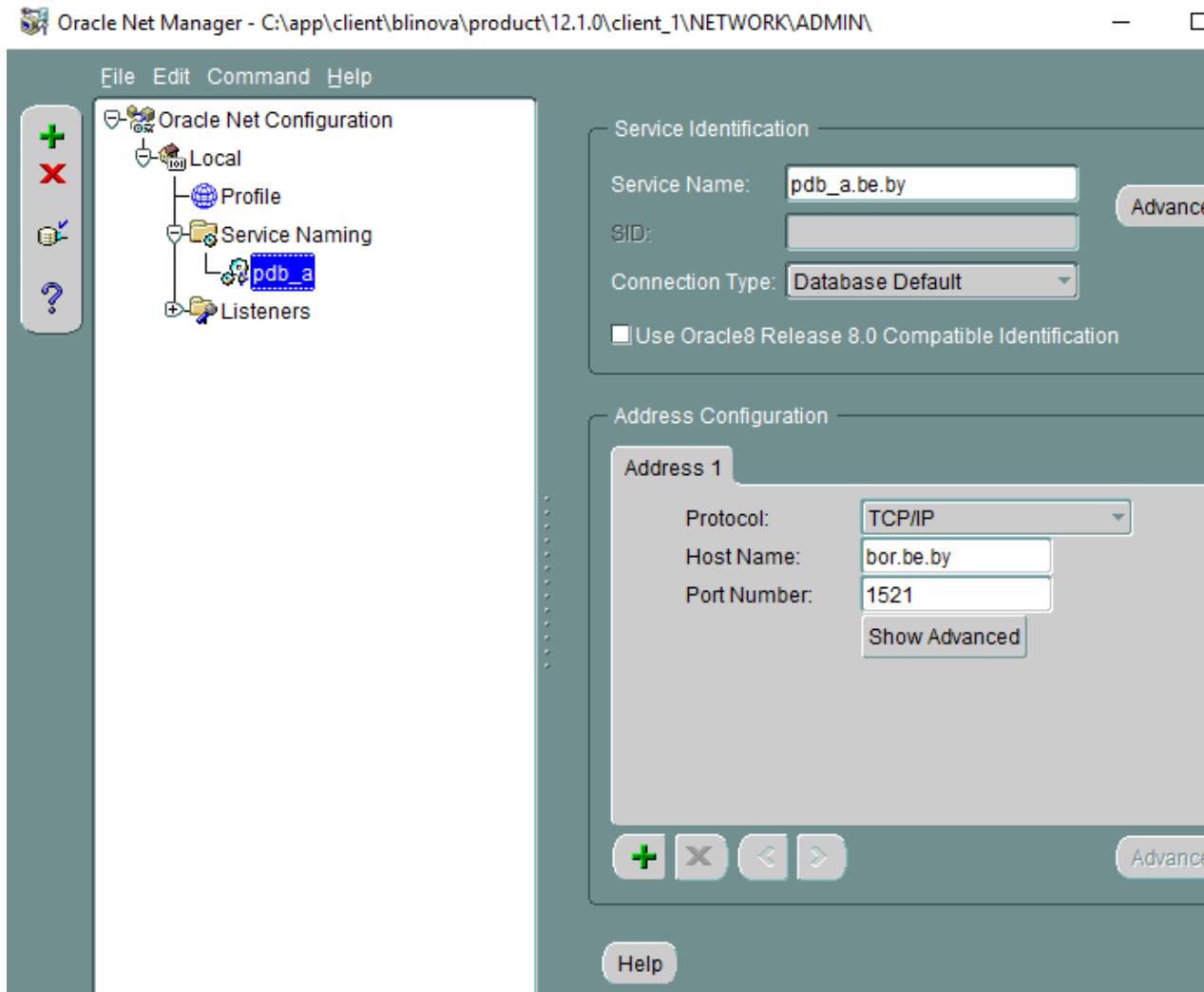
tnsnames.ora — Блокнот

Файл Правка Формат Вид Справка

```
# tnsnames.ora Network Configuration File: C:\app\client\blinova\product\12.1.0\client_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

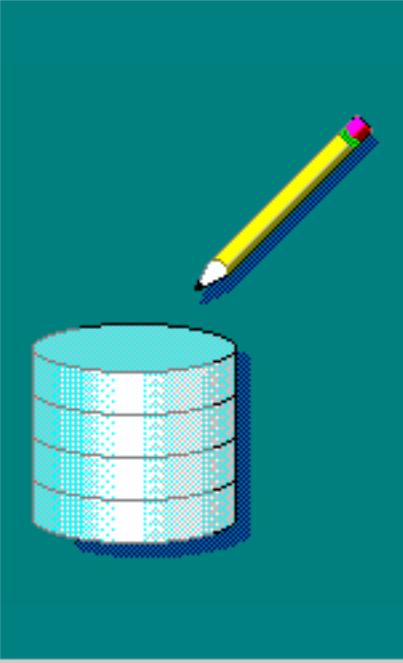
PDB_A =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bor.be.by)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdb_a.be.by)
    )
  )
```

# Создание локального именования



# Создание локального именования

Net Service Name Wizard: Welcome X



To access an Oracle database, or other service, across the network you use a net service name. This wizard will help you create a net service name.

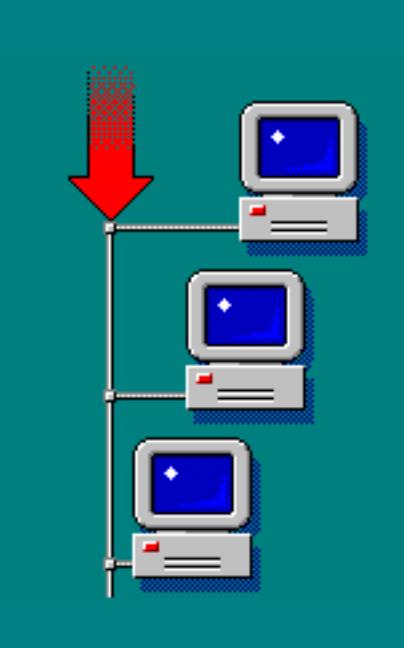
Enter the name you want to use to access the database or service. It can be any name you choose.

Net Service Name:   

Отмена Назад Следующий >

# Создание локального именования

Net Service Name Wizard, page 2 of 5: Protocol X



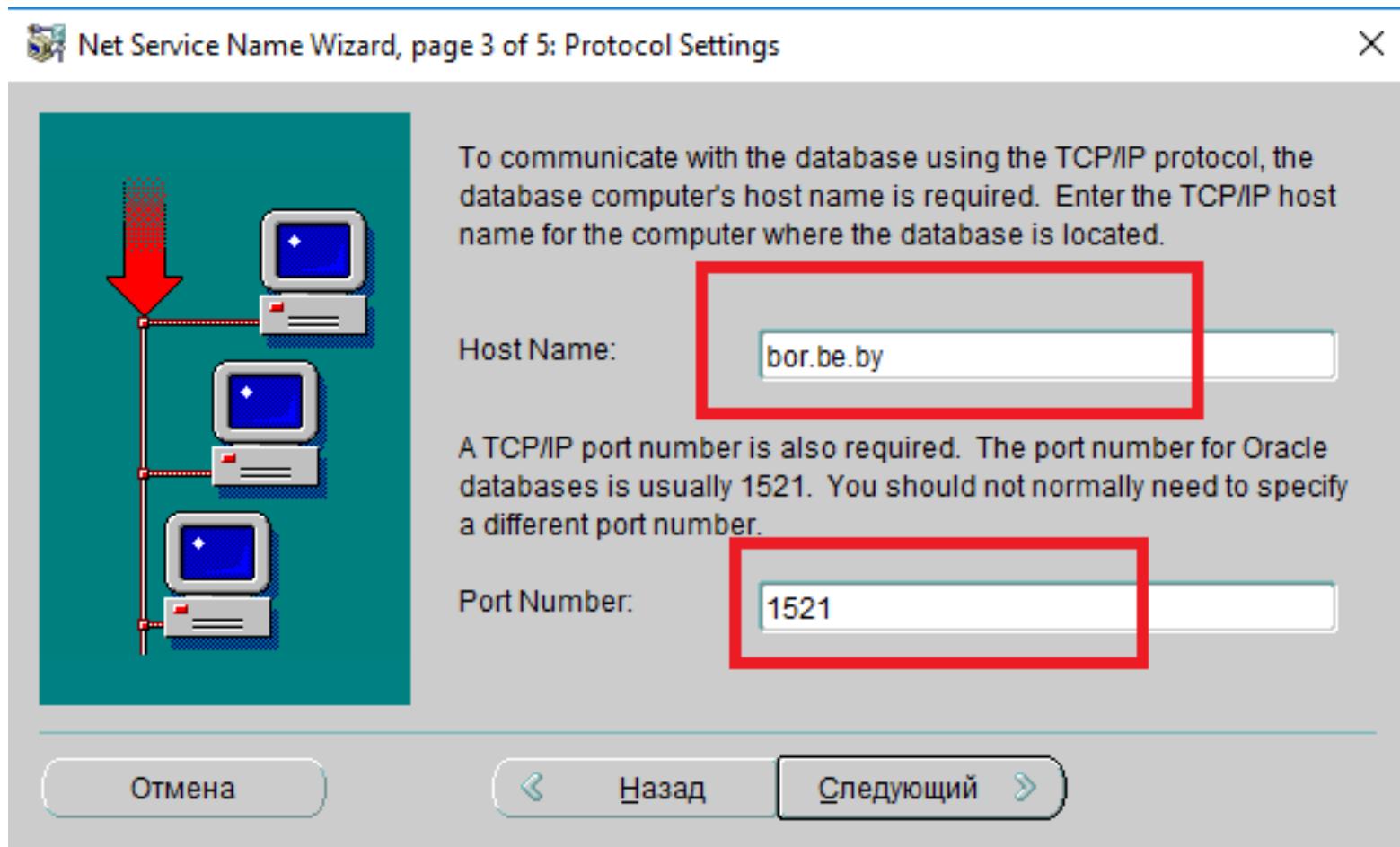
To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

**TCP/IP (Internet Protocol)**

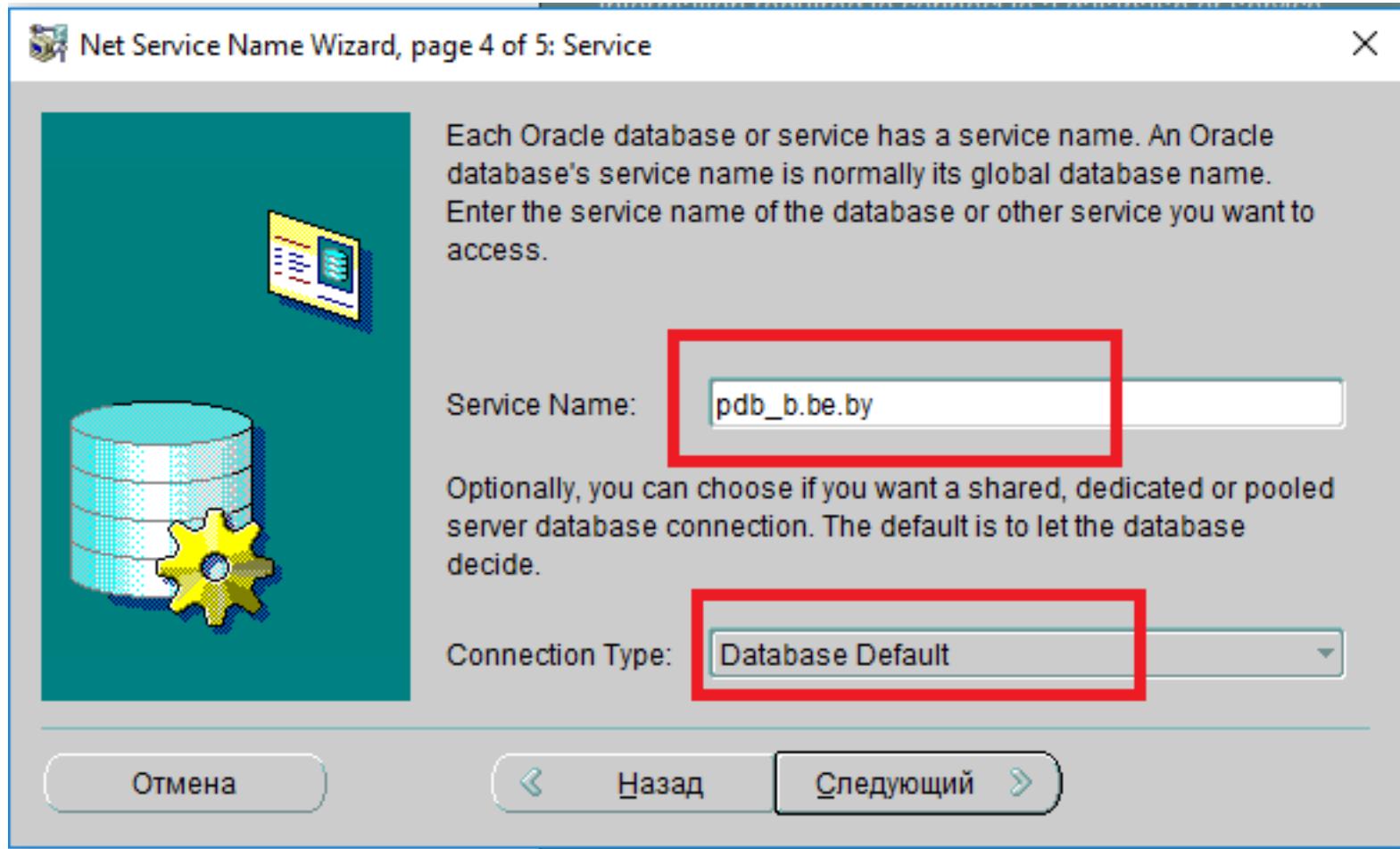
TCP/IP with SSL (Secure Internet Protocol)  
Named Pipes (Microsoft Networking)  
IPC (Local Database)

Отмена < Назад Следующий >

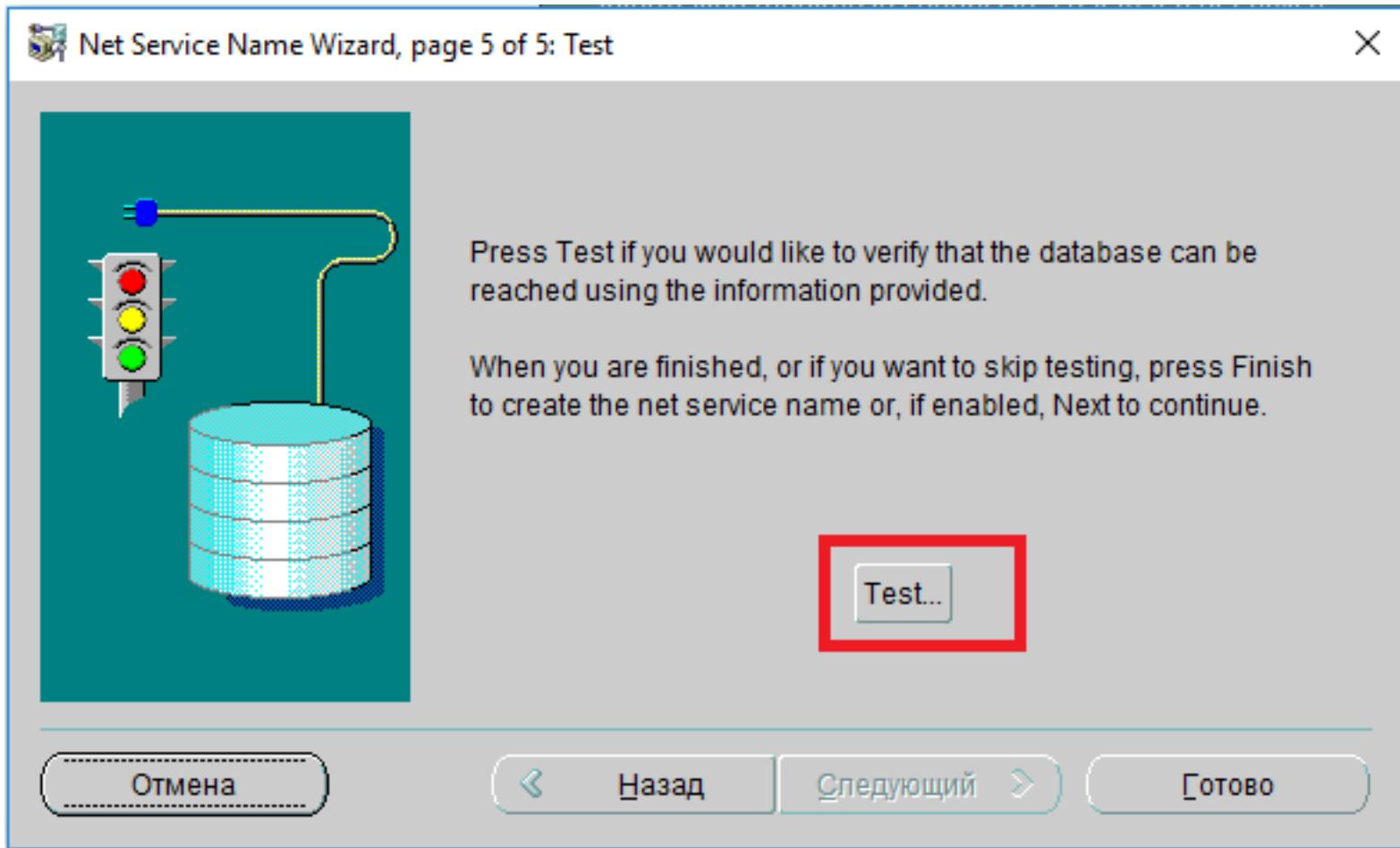
# Создание локального именования



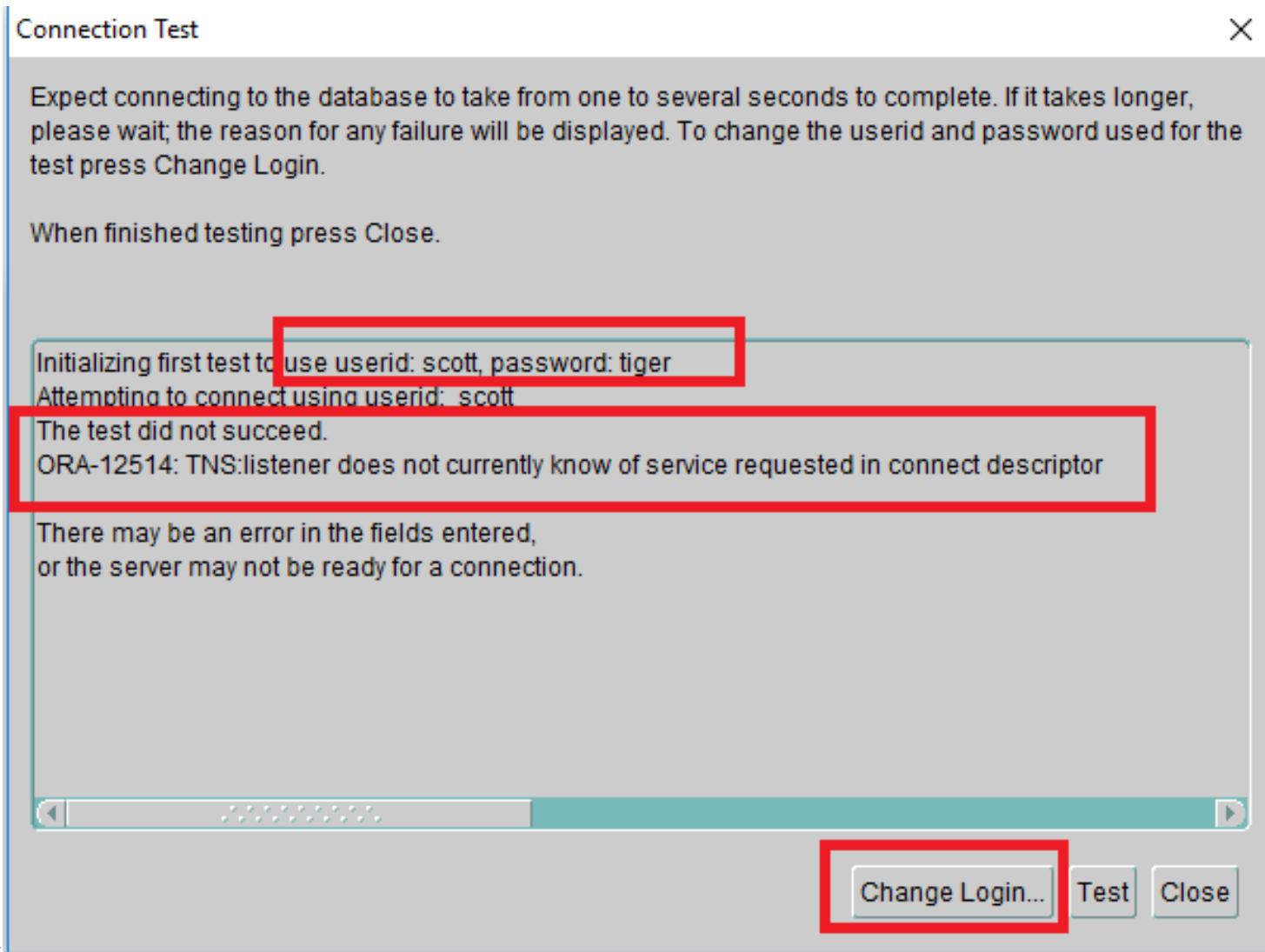
# Создание локального именования



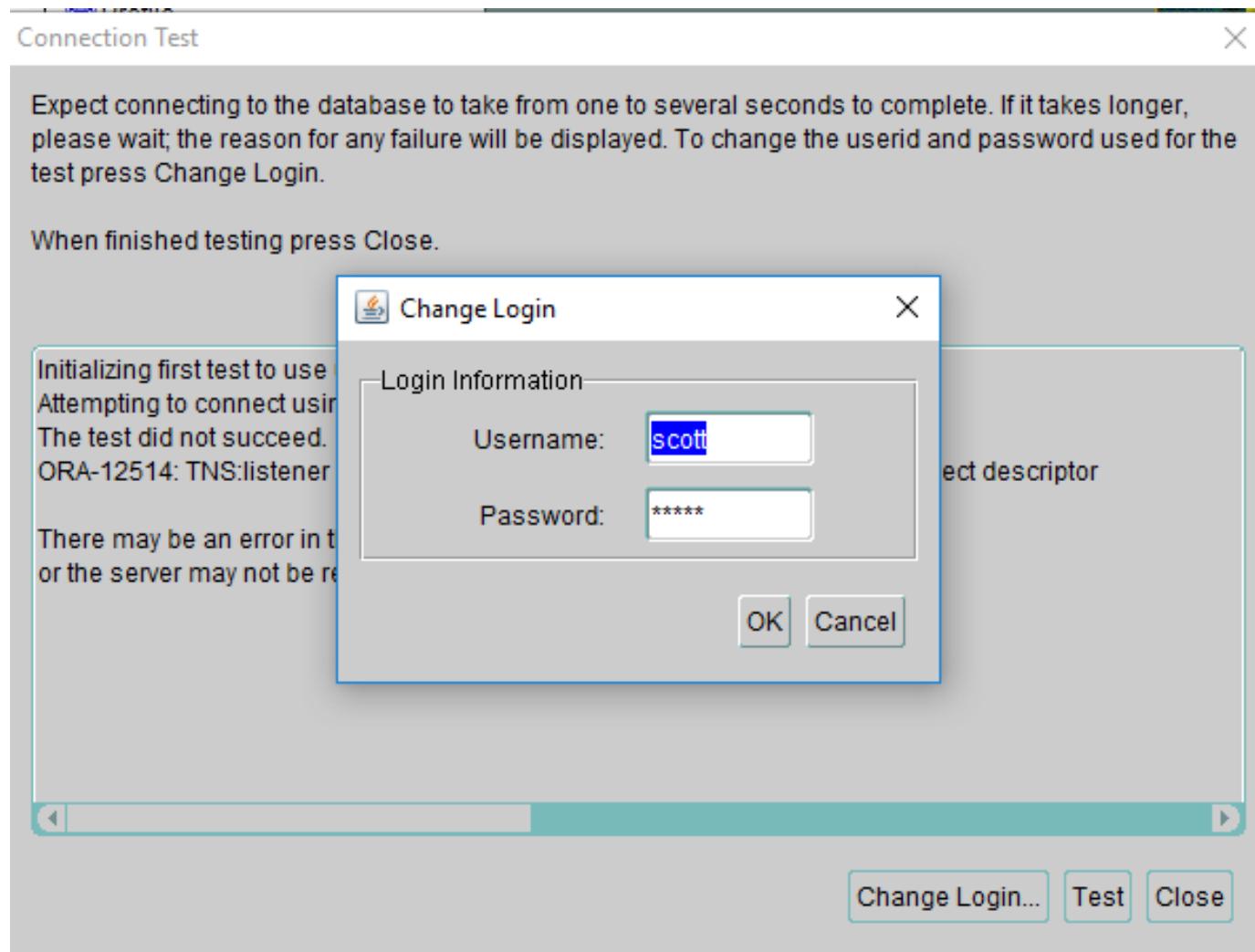
# Создание локального именования



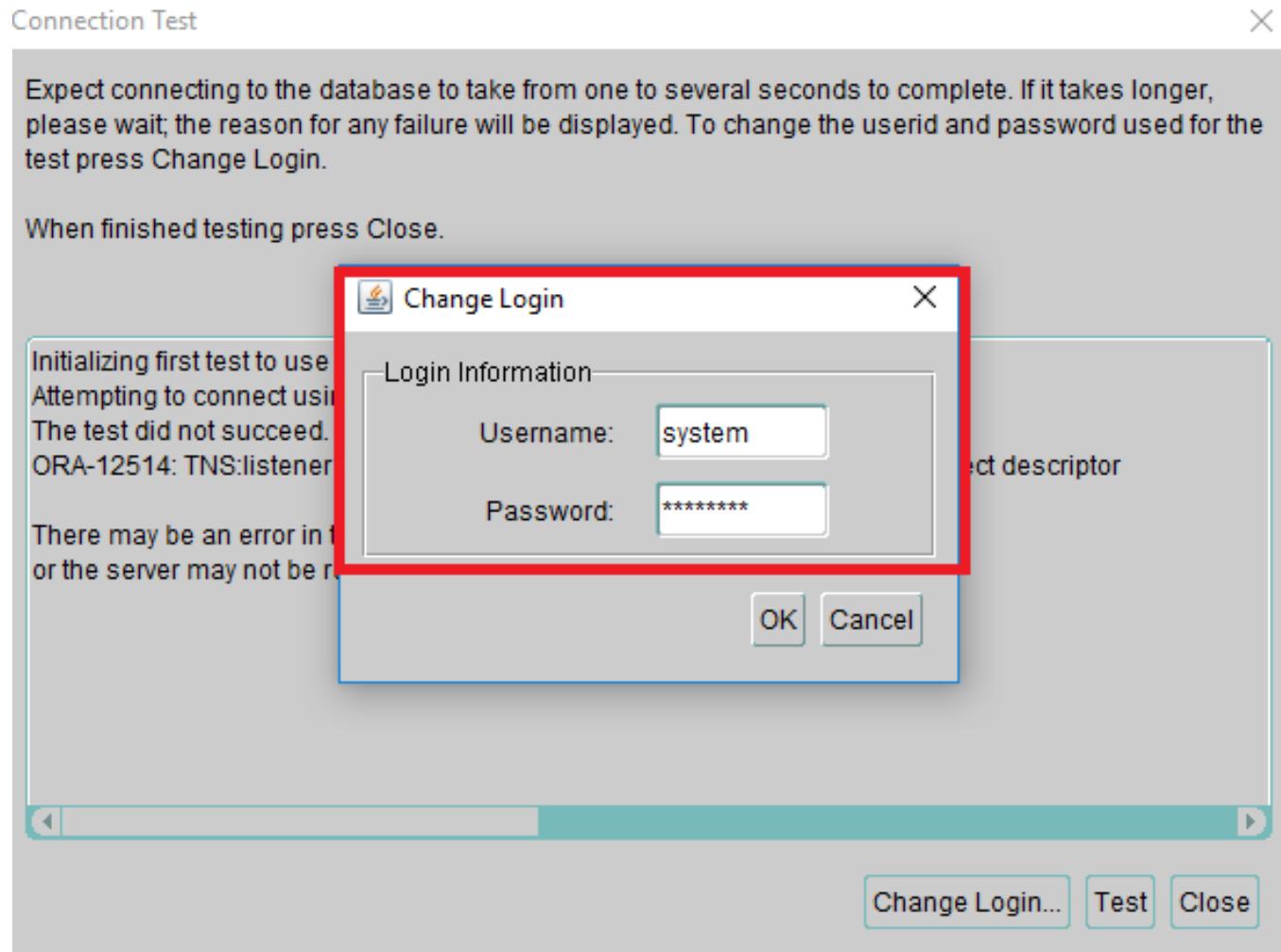
# Создание локального именования



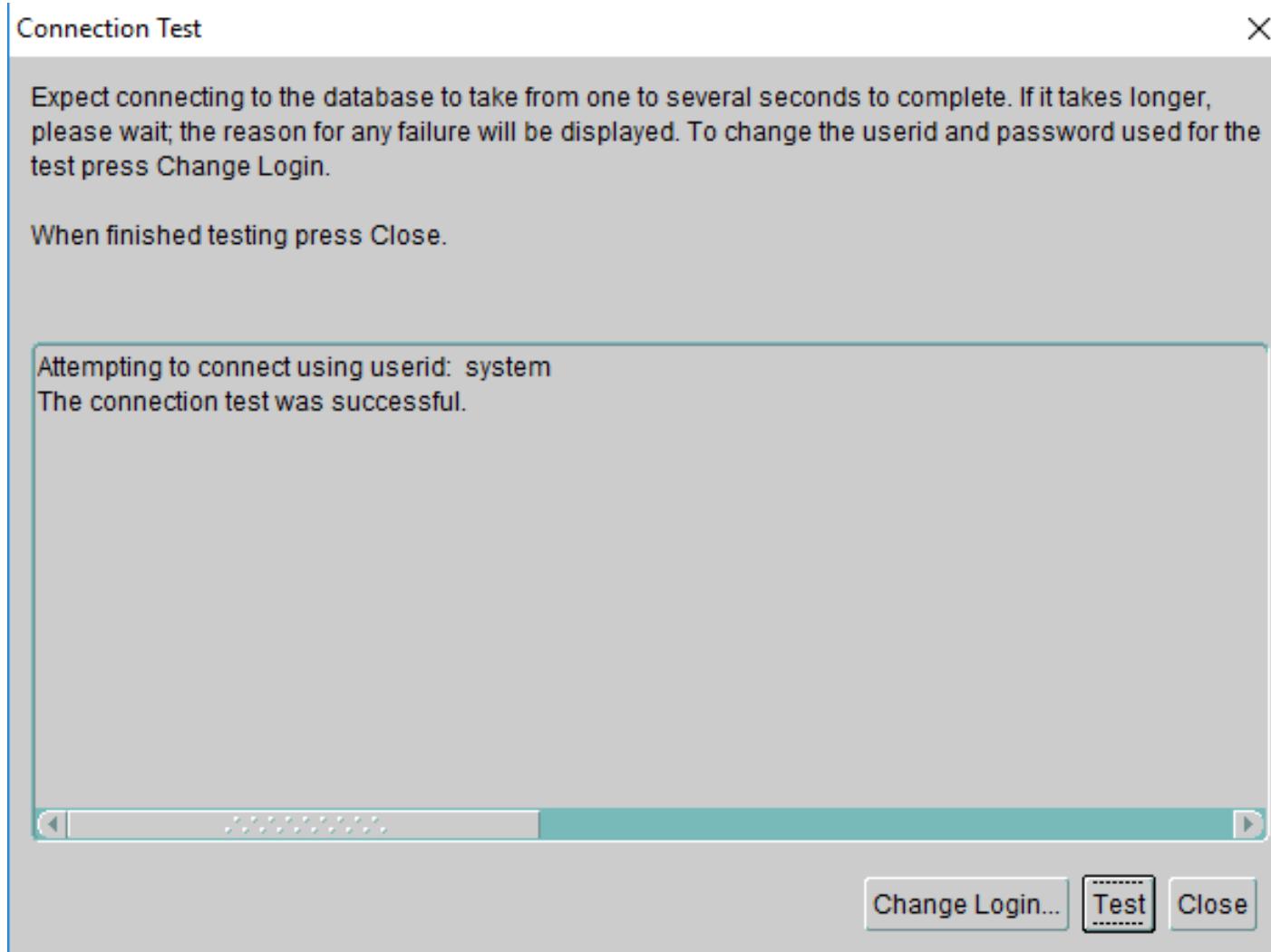
# Создание локального именования



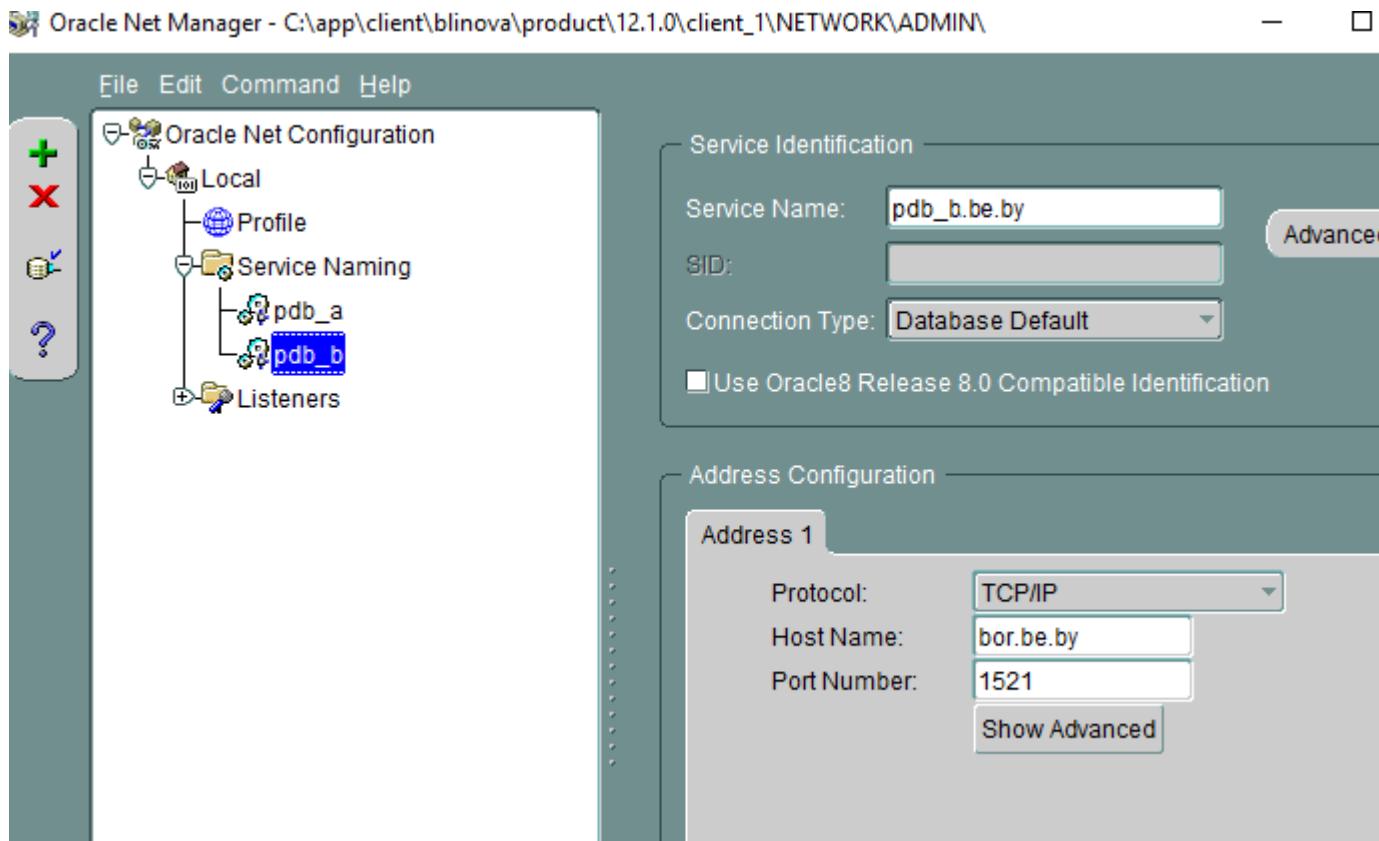
# Создание локального именования



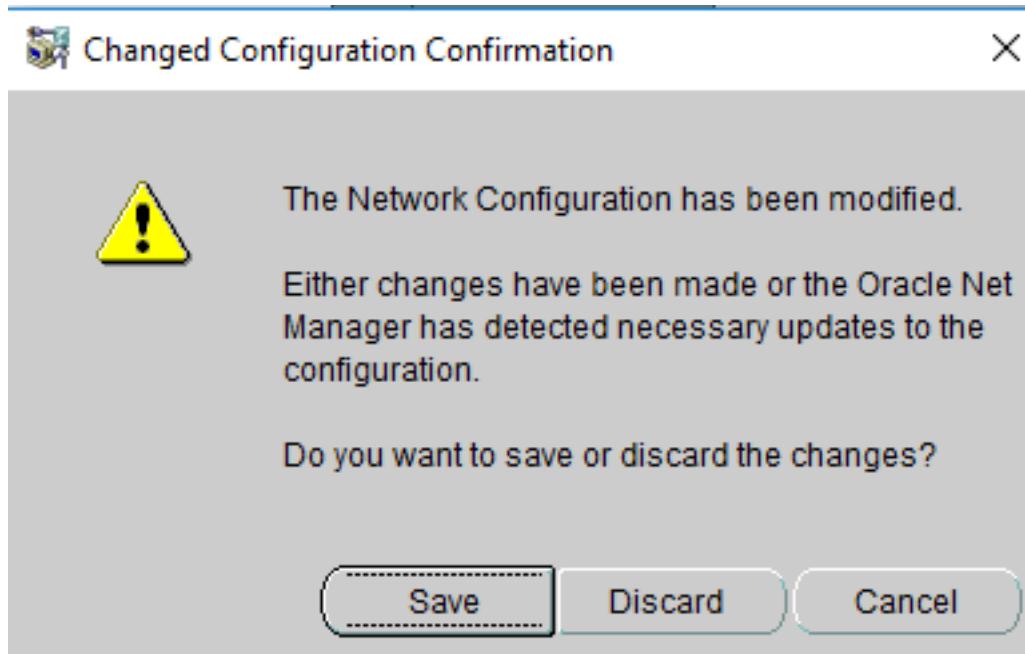
# Создание локального именования



# Создание локального именования



# Создание локального именования



# Создание локального именования

tnsnames.ora — Блокнот

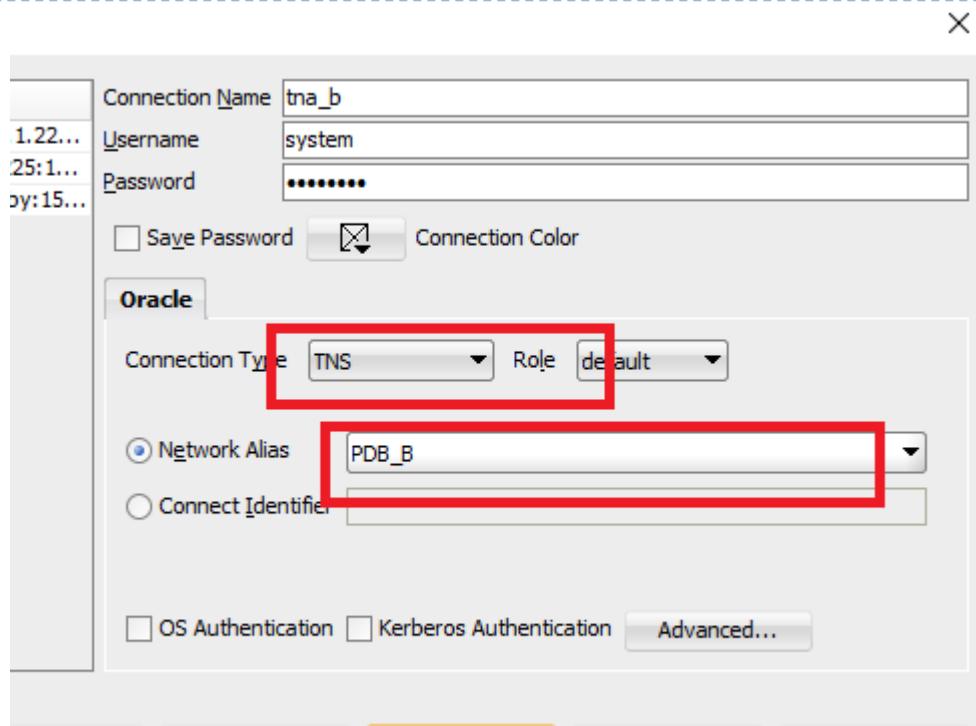
Файл Правка Формат Вид Справка

```
# tnsnames.ora Network Configuration File: C:\app\client\blinova\product\12.1.0\client_1\NETWORK\ADMIN\tns
# Generated by Oracle configuration tools.

PDB_B =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bor.be.by)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdb_b.be.by)
    )
  )

PDB_A =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = bor.be.by)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = pdb_a.be.by)
    )
  )
```

# Проверка TNS



```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Oct 25 18:16:02 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> connect system/Pa$$w0rd@pdb_b
Connected.
SQL>
```

# Утилита TNSPing

```
C:\Users\blinova>tnsping pdb_b

TNS Ping Utility for 64-bit Windows: Version 12.1.0.2.0 - Production on 25-OCT-2016 18:33:41
Copyright (c) 1997, 2014, Oracle. All rights reserved.

Used parameter files:
C:\app\client\blinova\product\12.1.0\client_1\network\admin\sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = bor.be.by)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME = pdb_b.be.by)))
OK (40 msec)
```



# LDAP-соединение

---

- ▶ Метод именования с помощью службы каталогов
- ▶ OID – Oracle Internet Directory
- ▶ LDAP – Lightweight Directory Access Protocol
- ▶ Требуется наличие специального LDAP-сервера



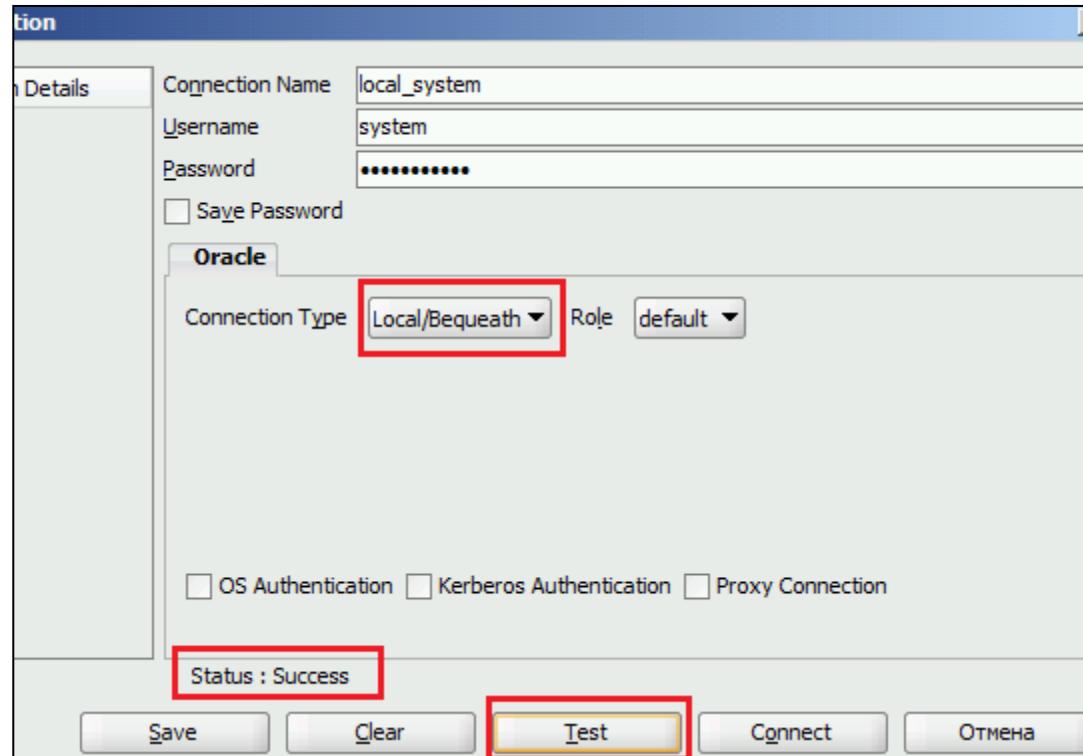
# Local/bequeath-соединение

---

- ▶ Только на сервере
- ▶ Можно соединяться с помощью `sqlplus` или `sqldeveloper`
- ▶ без указания параметров соединения
- ▶ только с выделенным сервером
- ▶ Listener не задействован
- ▶ Соединение со стандартным сервисом `SYSS$USERS`



# Local/bequeath-соединение



# Local/bequeath-соединение

Network Location Awareness	Collects an...	Running	Automatic	Network S...
Network Store Interface Service	This service ...	Running	Automatic	Local Service
Optimize drives	Helps the c...		Manual	Local Syste...
OracleJobSchedulerORCL			Disabled	\ora_instal...
OracleOraDB12Home1TNSRecoveryService		Running	Automatic	\ora_instal...
OracleOraDB12Home1TNSListener			Automatic	\ora_instal...
OracleRemExecServiceV2			Manual	Local Syste...
OracleServiceORCL		Running	Automatic	\ora_instal...
OracleVssWriterORCL		Running	Automatic	\ora_instal...

```
C:\app\ora_install_user\product\12.1.0\dbhome_1\BIN\sqlplus.exe

SQL*Plus: Release 12.1.0.1.0 Production on Tue Oct 25 08:30:20 2016
Copyright (c) 1982, 2013, Oracle. All rights reserved.

SQL> connect system/Pa$$w0rd@localhost:1521/orcl
ERROR:
ORA-12541: TNS:no listener

SQL> connect system/Pa$$w0rd
Connected.
SQL> _
```

# Вопросы?

---



# **Администрирование баз данных и приложений**

**Объекты базы данных**

**Лекция 8**

# Основные объекты базы данных

---

- ▶ Пользователи и схемы
  - ▶ Привилегии и роли
  - ▶ Таблицы, столбцы, ограничения и типы данных (в том числе абстрактные типы данных)
  - ▶ Последовательности
  - ▶ Кластеры и хэш-кластеры
  - ▶ Индексы
  - ▶ Синонимы
  - ▶ Представления
  - ▶ Моментальные снимки и материализованные представления
  - ▶ Связи баз данных
  - ▶ Секции
  - ▶ Процедуры, функции, пакеты
  - ▶ Триггеры
-

# Пользователи и схемы

---

- ▶ Учетная запись пользователя не является физической структурой
- ▶ Пользователям принадлежат объекты
- ▶ Схема – набор объектов, принадлежащий учетной записи пользователя
- ▶ Объекты создаются с правами учетных записей пользователей
- ▶ Учетные записи пользователей можно связать с учетными записями в ОС
- ▶ Двухкомпонентные имена – имя схемы.имя объекта



# Таблицы

---

- ▶ Таблица – основная структура сохранения информации в БД
- ▶ Типы таблиц:
  - ▶ Традиционные таблицы (heap organized table)
  - ▶ Индекс-таблицы (index organized table)
  - ▶ Кластеризованные индекс-таблицы (index clustered table)
  - ▶ Кластеризованные хэш-таблицы (hash clustered table)
  - ▶ Отсортированные кластеризованные хэш-таблицы (sorted hash clustered table)
  - ▶ Вложенные таблицы (nested table)
  - ▶ Временные таблицы (temporary table)
  - ▶ Объектные таблицы
  - ▶ Внешние таблицы



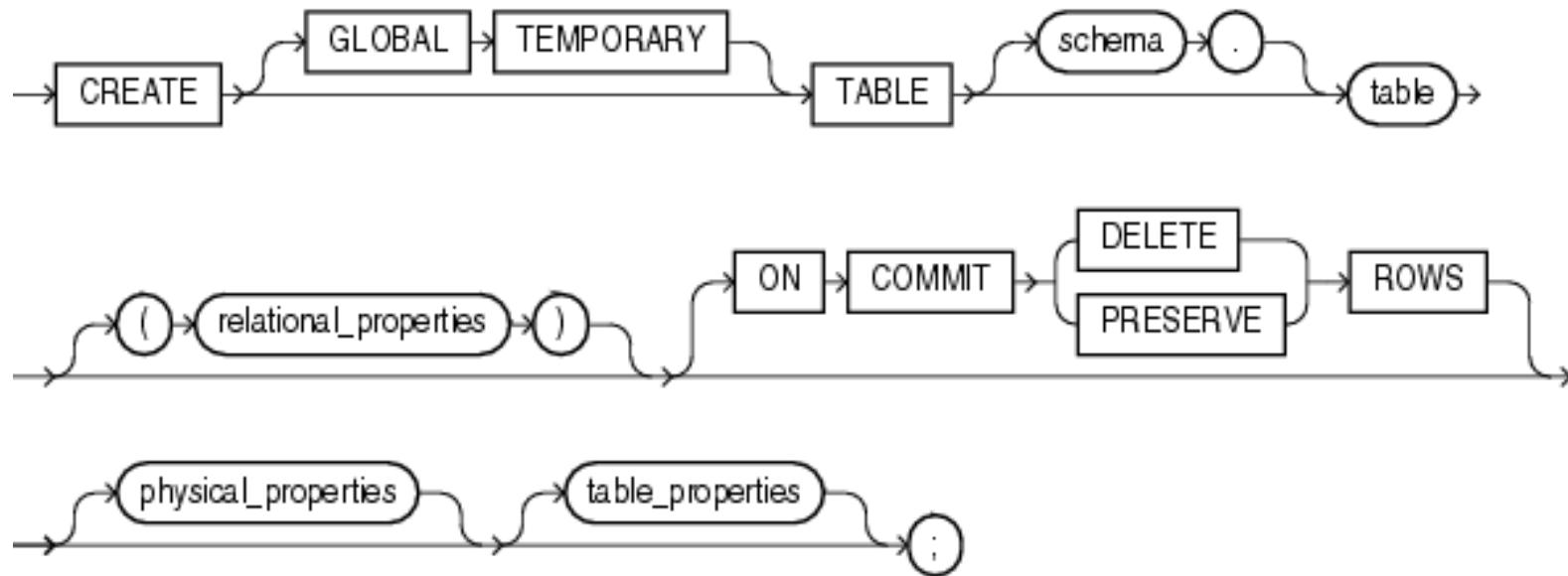
# Таблицы

---

- ▶ Может иметь до 1000 столбцов (<254)
- ▶ Может иметь неограниченное число строк
- ▶ Может иметь неограниченное число индексов
- ▶ Нет ограничения на число таблиц



# Таблицы



# Параметры

```
CREATE TABLE Employee(
    emp_id char(12),
    emp_name nvarchar2(100),
    hire_date date,
    CONSTRAINT pk_emp_id primary key (emp_id)
)
organization index
nomonitoring
logging
pctthreshold 20
```



# Параметры PCTFREE и PCTUSED

- ▶ Параметр PCTFREE – процент памяти блока, резервируемой для возможных обновлений строк, уже содержащихся в блоке
  - ▶ Параметр PCTUSED – процент занятой части памяти блока



## Управление параметрами заполнения блока

---

- ▶ Automatic segment space management - только PCTFREE
- ▶ Manual segment space management - PCTUSED,  
PCTTHRESHOLD, FREELISTS и др.



# Таблицы

---

- ▶ Типы данных:
  - ▶ CHAR / NCHAR
  - ▶ VARCHAR2 / NVARCHAR2
  - ▶ DATE
  - ▶ INTERVAL DAY TO SECOND / INTERVAL YEAR TO MONTH  
TIMESTAMP
  - ▶ TIMESTAMP WITH TIME ZONE / TIMESTAMP WITH LOCAL TIME
  - ▶ NUMBER (A,B)
  - ▶ LONG RAW/ LONG / RAW
  - ▶ BLOB / CLOB / NCLOB
  - ▶ ROWID / UROWID



# Типы данных ORACLE - символьные

CHAR	Символьное поле фиксированной длины до 2000 байт
NCHAR	Поле фиксированной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 2000 символов или 2000 байт в зависимости от набора символов.
VARCHAR2	Символьное поле переменной длины до 4000 байт
NVARCHAR2	Поле переменной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 4000 символов или 4000 байт в зависимости от набора символов.

# Поддержка национальных языков

---

- ▶ Переменная окружения **NLS\_LANG**:
- ▶ **NLS\_LANG = language\_territory.charset**
  - ▶ Язык (**LANGUAGE**) – имена месяцев, имена дней, направление текста, сокращения для времени и дат. По умолчанию **AMERICAN**
  - ▶ Территория (**TERRITORY**) – настройки календаря, формат даты, формат денежной единицы. Если не указан, то будет взято значение, соответствующее языку (для **RUSSIAN - CIS**)
  - ▶ Набор символов (**CHARACTER SET**) – отображение символов, отображение и конвертация заглавных букв, порядок замещения символов при преобразовании – кодовая страница



# Пример

 NLS_LANG	REG_SZ	AMERICAN_AMERICA.WE8MSWIN1252
---	--------	-------------------------------

- ▶ для Windows NLS-LANG может быть установлена в реестре и в переменных среды
- ▶ проверить, установлена ли переменная среды
  - ▶ HOST ECHO %NLS\_LANG%
- ▶ если ответ %NLS\_LANG%, то не установлена переменная среды имеет приоритет перед установкой в реестре
- ▶ рекомендуется устанавливать только в реестре

# Пример

```
SQL> SELECT * from NLS_INSTANCE_PARAMETERS;  
PARAMETER  
-----  
VALUE  
-----  
NLS_LANGUAGE  
AMERICAN  
  
NLS_TERRITORY  
AMERICA  
  
NLS_SORT
```

	PARAMETER	VALUE
1	NLS_LANGUAGE	AMERICAN
2	NLS_TERRITORY	AMERICA
3	NLS_SORT	(null)
4	NLS_DATE_LANGUAGE	(null)
5	NLS_DATE_FORMAT	(null)
6	NLS_CURRENCY	(null)
7	NLS_NUMERIC_CHARACTERS	(null)
8	NLS_ISO_CURRENCY	(null)
9	NLS_CALENDAR	(null)
10	NLS_TIME_FORMAT	(null)
11	NLS_TIMESTAMP_FORMAT	(null)
12	NLS_TIME_TZ_FORMAT	(null)
13	NLS_TIMESTAMP_TZ_FORMAT	(null)
14	NLS_DUAL_CURRENCY	(null)
15	NLS_COMP	BINARY
16	NLS_LENGTH_SEMANTICS	BYTE
17	NLS_NCHAR_CONV_EXCP	FALSE

```
SQL> SELECT * from NLS_DATABASE_PARAMETERS;
```

PARAMETER	VALUE
NLS_RDBMS_VERSION	12.1.0.1.0
NLS_NCHAR_CONV_EXCP	FALSE
NLS_LENGTH_SEMANTICS	BYTE
NLS_COMP	BINARY
NLS_DUAL_CURRENCY	\$
NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_FORMAT	HH.MI.SSXFF AM
NLS_SORT	BINARY
NLS_DATE_LANGUAGE	AMERICAN
NLS_DATE_FORMAT	DD-MON-RR
NLS_CALENDAR	GREGORIAN
NLS_NUMERIC_CHARACTERS	.,
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_CHARACTERSET	CL8MSWIN1251
NLS_ISO_CURRENCY	AMERICA
NLS_CURRENCY	\$
NLS_TERRITORY	AMERICA
NLS_LANGUAGE	AMERICAN

```
20 rows selected.
```

PARAMETER	VALUE
1 NLS_RDBMS_VERSION	12.1.0.1.0
2 NLS_NCHAR_CONV_EXCP	FALSE
3 NLS_LENGTH_SEMANTICS	BYTE
4 NLS_COMP	BINARY
5 NLS_DUAL_CURRENCY	\$
6 NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
7 NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
8 NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
9 NLS_TIME_FORMAT	HH.MI.SSXFF AM
10 NLS_SORT	BINARY
11 NLS_DATE_LANGUAGE	AMERICAN
12 NLS_DATE_FORMAT	DD-MON-RR
13 NLS_CALENDAR	GREGORIAN
14 NLS_NUMERIC_CHARACTERS	.,
15 NLS_NCHAR_CHARACTERSET	AL16UTF16
16 NLS_CHARACTERSET	CL8MSWIN1251
17 NLS_ISO_CURRENCY	AMERICA
18 NLS_CURRENCY	\$
19 NLS_TERRITORY	AMERICA
20 NLS_LANGUAGE	AMERICAN

# Пример

```
SQL> SELECT * FROM NLS_SESSION_PARAMETERS;  
-----  
PARAMETER          VALUE  
-----  
NLS_LANGUAGE        AMERICAN  
NLS_TERRITORY       AMERICA  
NLS_CURRENCY         $  
NLS_ISO_CURRENCY    AMERICA  
NLS_NUMERIC_CHARACTERS ,  
NLS_CALENDAR         GREGORIAN  
NLS_DATE_FORMAT     DD-MON-RR  
NLS_DATE_LANGUAGE    AMERICAN  
NLS_SORT             BINARY  
NLS_TIME_FORMAT      HH.MI.SSXFF AM  
NLS_TIMESTAMP_FORMAT DD-MON-RR HH.MI.SSXFF AM  
  
PARAMETER          VALUE  
-----  
NLS_TIME_TZ_FORMAT  HH.MI.SSXFF AM TZR  
NLS_TIMESTAMP_TZ_FORMAT DD-MON-RR HH.MI.SSXFF AM TZR  
NLS_DUAL_CURRENCY    $  
NLS_COMP              BINARY  
NLS_LENGTH_SEMANTICS BYTE  
NLS_NCHAR_CONV_EXCP FALSE  
  
17 rows selected.
```

PARAMETER	VALUE
1 NLS_LANGUAGE	RUSSIAN
2 NLS_TERRITORY	RUSSIA
3 NLS_CURRENCY	p.
4 NLS_ISO_CURRENCY	RUSSIA
5 NLS_NUMERIC_CHARACTERS	,
6 NLS_CALENDAR	GREGORIAN
7 NLS_DATE_FORMAT	DD/MM/YY
8 NLS_DATE_LANGUAGE	RUSSIAN
9 NLS_SORT	RUSSIAN
10 NLS_TIME_FORMAT	HH24:MI:SSXFF
11 NLS_TIMESTAMP_FORMAT	DD.MM.RR HH24:MI:SSXFF
12 NLS_TIME_TZ_FORMAT	HH24:MI:SSXFF TZR
13 NLS_TIMESTAMP_TZ_FORMAT	DD.MM.RR HH24:MI:SSXFF TZR
14 NLS_DUAL_CURRENCY	p.
15 NLS_COMP	BINARY
16 NLS_LENGTH_SEMANTICS	BYTE
17 NLS_NCHAR_CONV_EXCP	FALSE

# Пример

```
create table test (c char);
insert into test (c) values ('A');
commit;
select c, ascii(c), ascii(str(c) from test;
```

```
SQL> insert into test (c) values ('A');
1 row created.

SQL> insert into test (c) values ('a');
1 row created.

SQL> insert into test (c) values ('к');
1 row created.

SQL> commit;
```

01/01/2010 00:00:00

	C	ASCII(C)	ASCIISTR(C)
1	c	99	c
2	к	234 \043A	
3	a	224 \0430	
4	A	192 \0410	

	C	ASCII(C)	ASCIISTR(C)
1	c	99	c
2	к	234 \043A	
3	a	224 \0430	
4	A	192 \0410	
5	€	136 \20AC	
6		160 \00AO	
7	?	63	?

```
SQL> select * from test;
C
-
c
?
?
?
A
a
?

7 rows selected.
```

# NLS

---

- ▶ **NLS\_DATABASE\_PARAMETERS** – устанавливается при создании БД
- ▶ **NLS\_INSTANCE\_PARAMETERS** – устанавливается для экземпляра
- ▶ **NLS\_SESSION\_PARAMETERS** – устанавливается для сессии

```
-- https://docs.oracle.com/cd/A83908_02/NT816EE/DOC/nt.816/a73010/apb.htm
-- NLS_LANG = LANGUAGE_TERRITORY.CHARACTER_SET
-- AMERICAN_AMERICA.WE8MSWIN1252
```



128	Б	144	б	160		176	*	192	А	208	Р	224	а	240	р
129	Ѓ	145	'	161	Ў	177	±	193	Б	209	С	225	б	241	с
130	,	146	'	162	ў	178	І	194	В	210	Т	226	в	242	т
131	ѓ	147	"	163	Ј	179	і	195	Г	211	У	227	г	243	у
132	"	148	"	164	Ѡ	180	Ր	196	Д	212	Փ	228	դ	244	ֆ
133	...	149	*	165	Ӯ	181	Ը	197	Ե	213	Խ	229	ե	245	խ
134	†	150	-	166	՚	182	՚	198	Ժ	214	Ծ	230	ժ	246	ց
135	‡	151	-	167	Ը	183	.	199	Յ	215	Ч	231	չ	247	չ
136	.	152	.	168	Ӭ	184	ё	200	И	216	Ш	232	и	248	շ
137	%	153	™	169	©	185	№	201	Ӣ	217	Щ	233	й	249	щ
138	Љ	154	љ	170	Ը	186	Ը	202	Կ	218	՚	234	կ	250	՚
139	<	155	,	171	«	187	»	203	Լ	219	՚	235	լ	251	՚
140	Њ	156	њ	172	՝	188	յ	204	Մ	220	՚	236	մ	252	՚
141	Ќ	157	ќ	173	-	189	Տ	205	Ն	221	Э	237	ն	253	է
142	Ћ	158	ћ	174	®	190	Ը	206	Օ	222	Յ	238	օ	254	յ
143	Џ	159	џ	175	՚	191	՚	207	Պ	223	Յ	239	պ	255	յ

	C	ASCII(C)	ASCIISTR(C)
1	с	99	с
2	к	234 \043А	
3	а	224 \0430	
4	А	192 \0410	
5	€	136 \20AC	
6		160 \00A0	
7	?	63	?

SQL> select \* from test;

С  
-  
с  
Г  
г  
Ա  
ա  
?

7 rows selected.

# Пример

```
select c, ascii(c), asciistr(c), DUMP(c, 1010) from test;
```

	C	ASCII(C)	ASCIISTR(C)	DUMP(C,1010)
1	с	99	с	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 99
2	к	234	\043A	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 234
3	а	224	\0430	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 224
4	А	192	\0410	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 192
5	€	136	\20AC	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 136
6		160	\00A0	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 160
7	?	63	?	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 63

# Типы данных ORACLE – символьные

LONG	Символьный, переменной длины, до 2GB, оставлен для совместимости
RAW(n)	Переменной длины, для бинарных данных $n \leq 2000$ byte оставлен для совместимости
LONG RAW	Бинарные данные до 2GB
CLOB	Символьный тип большой объект до 4GB
NLOB	CLOB для многобайтных символов
BLOB	Большой двоичный объект до 4GB
BFILE	Указатель на двоичный файл операционной системы



# Типы данных ORACLE – дата/время

DATE	7 байтовое поле фиксированной длины, используемое для хранения даты и времени
INTERVAL DAY TO SECOND	11 байтовое поле фиксированной длины для интервала времени: Дни, часы, минуты, секунды
INTERVAL YEAR TO MONTH TIMESTAMP	5 байтовое поле фиксированной длины для интервала времени: Годы и месяцы
TIMESTAMP WITH TIME ZONE	13 байтовое поле фиксированной длины Дата, время и настройки, связанные с часовым поясом.
TIMESTAMP WITH LOCAL TIME	7-11 байтовое поле переменной длины Дата и время, приведенные к часовому поясу базы данных



# Типы данных ORACLE – числовые

NUMBER(n, s)

Числовой тип переменной длины

Точность n <= 38, общее количество цифр

Масштаб s = [-84,127], количество цифр после запятой

# Таблицы

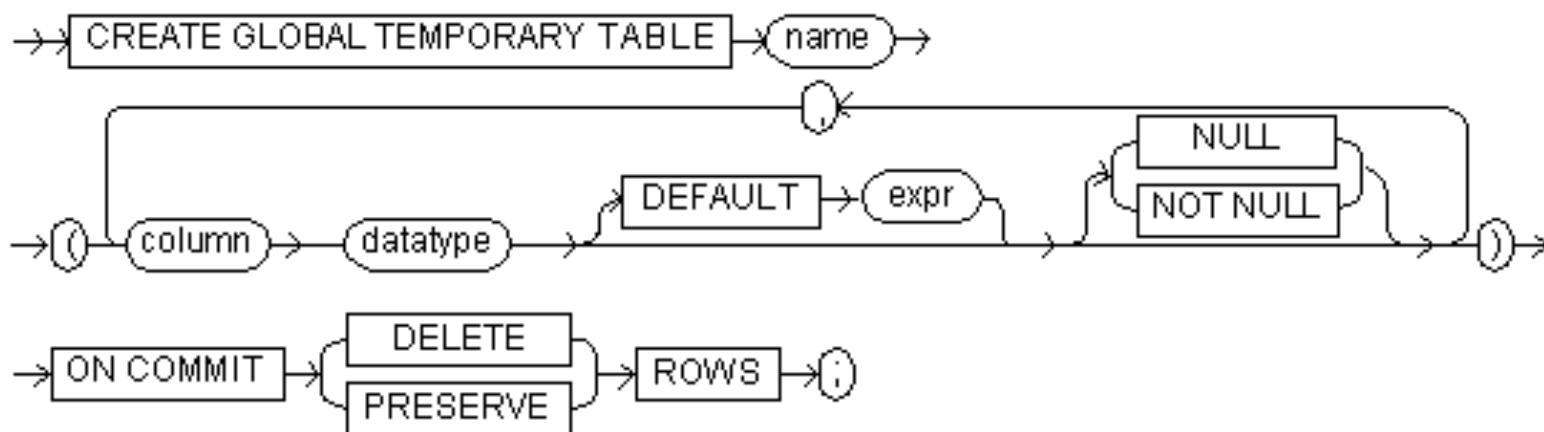
---

- ▶ ROWID – 16-тиричный тип для уникального определения любой строки любой таблицы в БД
- ▶ Длина – 18 символов, которая делится на 4 элемента (6 + 3 + 6 + 3):
  - ▶ ОООООО – уникальный номер объекта в БД, которому принадлежит строка;
  - ▶ FFF – уникальный номер файла данных БД, где хранится строка;
  - ▶ BBBBВВ – номер блока данных, который хранит строку, уникален на уровне файла данных БД;
  - ▶ RRR – адрес строки в блоке
- ▶ UROWID – специальный 16-тиричный тип данных для адресации строк в таблицах, организованных по индексу



## Временные таблицы

- ▶ Временные таблицы – механизм хранения данных в БД
  - ▶ Состоит из столбцов и строк, как и обычная таблица



# Временные таблицы

---

- ▶ Временные таблицы – глобальны
- ▶ Привилегии для создания временной таблицы  
**CREATE TABLE**
- ▶ Можно разместить временную таблицу в заданном табличном пространстве.
- ▶ Временные таблицы – это шаблон, хранящийся в словаре базы данных, для нее выделяется временный сегмент в (по умолчанию) TEMPORARY-табличном пространстве и для каждого пользователя свой.
- ▶ Каждый пользователь видит только свои данные (свой сегмент данных).



# Временные таблицы

---

- ▶ Статичны: временные таблицы создаются (`CREATE`) один раз и существуют, пока их не удалят (`DROP`).
- ▶ `DROP` не получится, если таблица в этот момент используется другим пользователем.



# Временные таблицы

---

- ▶ Временные таблицы бывают:
  - ▶ ON COMMIT PRESERVE ROWS – на время сеанса, данные существуют только на время сеанса, возможны все DML-операторы, TCL-операторы
  - ▶ ON COMMIT DELETE ROWS – на время транзакции, данные существуют только на время транзакции, возможны все DML-операторы, после выполнения COMMIT или ROLLBACK таблица становится пустой
- ▶ В начале сеанса временная таблица всегда пуста



# Временные таблицы

---

- ▶ Для временных таблиц можно создавать триггеры
- ▶ Для временных таблиц можно указать констрайны (ограничения)
- ▶ Для временных таблиц можно создавать индексы



# Временные таблицы

---

- ▶ Не могут быть индексно-организованными, нельзя секционировать, размещать в кластере.
- ▶ Данные повторного выполнения генерируются, но их количество пренебрежительно мало.



# Временные таблицы

```
-- 15/01.sql
```

```
create global temporary table gtemp_tlesson
(
    tlesson      char(10 byte) not null,
    tlesson_name varchar2(30 byte),
    constraint pk_temp_tlesson primary key (tlesson)
)

insert into gtemp_tlesson (tlesson, tlesson name)
(
    select tlesson, tlesson name
    from tlesson
)

select * from gtemp_tlesson

drop table gtemp_tlesson;
```



# Временные таблицы

```
-- 15/02.sql

create global temporary table gtemp_tlesson as select * from tlesson;
insert into gtemp_tlesson (tlesson, tlesson_name)
(
    select tlesson, tlesson_name
    from tlesson
)
select * from gtemp_tlesson
drop table gtemp_tlesson;
```



# Временные таблицы

```
-- 15/05.sql
create global temporary table gtemp_tlesson on commit preserve rows
    as select * from tlesson

insert into gtemp_tlesson ( select * from tlesson) ~

~~~~
select * from gtemp_tlesson
delete gtemp_tlesson
commit
drop table gtemp tlesson
```

Error starting at line 10 in command:

```
drop table gtemp_tlesson
```

Error report:

SQL Error: ORA-14452: попытка создать, изменить или удалить индекс в уже используемой  
14452. 00000 - "attempt to create, alter or drop an index on temporary table already

\*Cause: An attempt was made to create, alter or drop an index on temporary  
table which is already in use.

\*Action: All the sessions using the session-specific temporary table have  
to truncate table and all the transactions using transaction  
specific temporary table have to end their transactions.

# Временные таблицы

```
-- 15/04.sql
create global temporary table gtemp_tlesson on commit delete rows
    as select * from tlesson

insert into gtemp_tlesson ( select * from tlesson)

select * from gtemp_tlesson
delete gtemp_tlesson
commit
drop table gtemp_tlesson
```

---

```
drop table gtemp_tlesson succeeded.
```

---



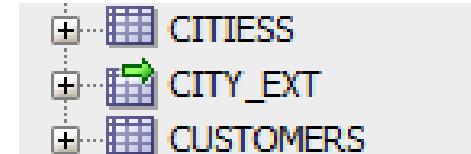
# EXTERNAL TABLE

	A	B	C	D	E	F	G	H	I	J
1	Malisheve	Malisheve	42.4822	20.7458	Kosovo	XK,XKS	Malisheve	admin,,	1901597212	
2	Prizren	Prizren	42.2139	20.7397	Kosovo	XK,XKS	Prizren	admin,,	1901360309	
3	Zubin Potok	Zubin Potok	42.9144	20.6897	Kosovo	XK,XKS	Zubin Potok	admin,,	1901608808	
4	Kamenice	Kamenice	42.5781	21.5803	Kosovo	XK,XKS	Kamenice	admin,,	1901851592	
5	Viti	Viti	42.3214	21.3583	Kosovo	XK,XKS	Viti	admin,,	1901328795	
6	Shterpce	Shterpce	42.2394	21.0272	Kosovo	XK,XKS	Shterpce	admin,,	1901828239	
7	Shtime	Shtime	42.4331	21.0397	Kosovo	XK,XKS	Shtime	admin,,	1901598505	
8	Vushtrri	Vushtrri	42.8231	20.9675	Kosovo	XK,XKS	Vushtrri	admin,,	1901107642	
9	Dragash	Dragash	42.0265	20.6533	Kosovo	XK,XKS	Dragash	admin,,	1901112530	
10	Podujeve	Podujeve	42.9111	21.1899	Kosovo	XK,XKS	Podujeve	admin,,	1901550082	
11	Fushe Kosove	Fushe Kosove	42.6639	21.0961	Kosovo	XK,XKS	Fushe Kosove	admin,,	1901134407	
12	Kacanik	Kacanik	42.2319	21.2594	Kosovo	XK,XKS	Kacanik	admin,,	1901200321	
13	Kline	Kline	42.6217	20.5778	Kosovo	XK,XKS	Kline	admin,,	1901230162	
14	Leposaviq	Leposaviq	43.1039	20.8028	Kosovo	XK,XKS	Leposaviq	admin,,	1901974597	
15	Peje	Peje	42.6600	20.2922	Kosovo	XK,XKS	Peje	admin,,	1901339694	
16	Rahovec	Rahovec	42.3994	20.6547	Kosovo	XK,XKS	Rahovec	admin,,	1901336358	
17	Gjilan	Gjilan	42.4689	21.4633	Kosovo	XK,XKS	Gjilan	admin,,	1901235642	
18	Lipjan	Lipjan	42.5217	21.1258	Kosovo	XK,XKS	Lipjan	admin,,	1901682048	

# EXTERNAL TABLE

```
-- ORACLE_LOADER type
create or replace directory EXT_DIR as 'd:\external_dir';

CREATE TABLE city_ext (
    city          VARCHAR2(50),
    city_ascii    VARCHAR2(50),
    lat           VARCHAR2(50),
    lng           VARCHAR2(50),
    country       VARCHAR2(50),
    iso2          VARCHAR2(50),
    iso3          VARCHAR2(50),
    admin_name    VARCHAR2(50),
    capital       VARCHAR2(50),
    population    VARCHAR2(50),
    id            VARCHAR2(50)
)
ORGANIZATION EXTERNAL (
    TYPE ORACLE_LOADER
    DEFAULT DIRECTORY EXT_DIR
    ACCESS PARAMETERS (
        RECORDS DELIMITED BY NEWLINE
        FIELDS TERMINATED BY ','
        MISSING FIELD VALUES ARE NULL)
    LOCATION ('data.csv')
)
PARALLEL 2 REJECT LIMIT UNLIMITED;
```



# EXTERNAL TABLE

SELECT \* FROM city\_ext;

Script Output X Query Result X

SQL | Fetched 50 rows in 0,652 seconds

	CITY	CITY_ASCII	LAT	LNG	COUNTRY	ISO2	ISO3	ADMIN_NAME	CAPITAL	POPULATION	ID
1	Malisheve	Malisheve	42.4822	20.7458	Kosovo	XK	XKS	Malisheve	admin	(null)	1901597212
2	Prizren	Prizren	42.2139	20.7397	Kosovo	XK	XKS	Prizren	admin	(null)	1901360309
3	Zubin Potok	Zubin Potok	42.9144	20.6897	Kosovo	XK	XKS	Zubin Potok	admin	(null)	1901608808
4	Kamenice	Kamenice	42.5781	21.5803	Kosovo	XK	XKS	Kamenice	admin	(null)	1901851592
5	Viti	Viti	42.3214	21.3583	Kosovo	XK	XKS	Viti	admin	(null)	1901328795
6	Shterpce	Shterpce	42.2394	21.0272	Kosovo	XK	XKS	Shterpce	admin	(null)	1901828239
7	Shtime	Shtime	42.4331	21.0397	Kosovo	XK	XKS	Shtime	admin	(null)	1901598505
8	Vushtrri	Vushtrri	42.8231	20.9675	Kosovo	XK	XKS	Vushtrri	admin	(null)	1901107642
9	Dragash	Dragash	42.0265	20.6533	Kosovo	XK	XKS	Dragash	admin	(null)	1901112530
10	Podujeve	Podujeve	42.9111	21.1899	Kosovo	XK	XKS	Podujeve	admin	(null)	1901550082
11	Rrëshë Kosovë	Rrëshë Kosovë	42.6630	21.0061	Kosovo	VV	VVK	Rrëshë Kosovë	admin	(null)	1901134407

# EXTERNAL TABLE

```
UPDATE city_ext SET city = 'MINSK';

Script Output X | Query Result X
✖️ ✎ 📁 📄 | Task completed in 0,114 seconds
```

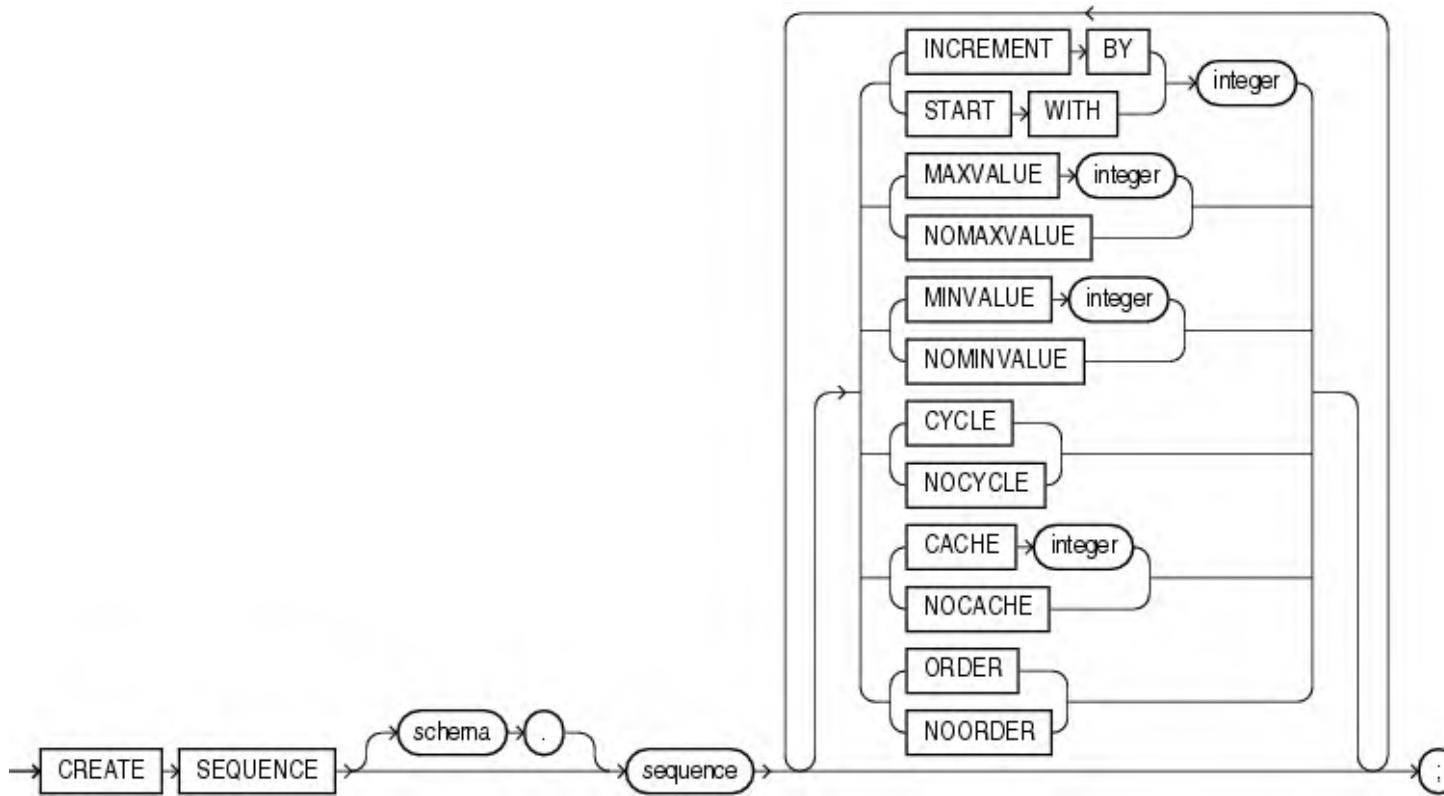
Directory EXT\_DIR created.

Table CITY\_EXT created.

Error starting at line : 31 in command -  
UPDATE city\_ext SET city = 'MINSK'  
Error at Command Line : 31 Column : 8  
Error report -  
SQL Error: ORA-30657: операция во внешне-организованной таблице не поддерживается  
30657.0000 - "operation not supported on external organized table"  
\*Cause: User attempted an operation on an external table which is  
not supported.  
\*Action: Don't do that!

# Последовательности

- ▶ Последовательность – объект базы данных, предназначенный для генерации числовой последовательности



# Последовательности

---

## ▶ Привилегия CREATE SEQUENCE

```
--  
SUVCORE@sh1> create sequence SUVCORE.SHEDULESSON_ID  
 2  increment by    10  
 3  start    with  100  
 4  nomaxvalue          -- maxvalue  
 5  nominvalue          -- minvalue  
 6  noscycle            -- cycle  
 7  cache 20             -- nocache  
 8  order                -- noorder  
 9 /
```

Последовательность создана.



# Последовательности

```
SUVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;
select SHEDULESSON_ID.CURRVAL FROM DUAL
*
ошибка в строке 1:
ORA-08002: последов. SHEDULESSON_ID.CURRVAL еще не определен в этом сеансе

SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL From dual;

NEXTVAL
-----
100

SUVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;

CURRVAL
-----
100

SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL From dual;

NEXTVAL
-----
110

SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL, SHEDULESSON_ID.NEXTVAL From dual;

NEXTVAL      NEXTVAL
-----      -----
120          120

SUVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;

CURRVAL
-----
120
```



# Последовательности

---

- ▶ Представления словаря:
  - ▶ SYS.DBA\_SEQUENCES
  - ▶ SYS.ALL\_SEQUENCES
  - ▶ SYS.USER\_SEQUENCES



# Oracle Identity

---

```
CREATE TABLE EMP (
    ID NUMBER GENERATED BY DEFAULT ON NULL
        AS IDENTITY START WITH 1000
        MINVALUE 1000
        MAXVALUE 10000
        INCREMENT BY 10
        CACHE 20,
    NAME VARCHAR2(100) NOT NULL,
    DEPT INT NOT NULL);
```



# Oracle Identity

---

- ▶ GENERATED ALWAYS AS IDENTITY — значения определяются только сервером
- ▶ GENERATED BY DEFAULT AS IDENTITY — позволяет указать значение. Если значение не указано, то сервер назначит значение из последовательности
- ▶ GENERATED BY DEFAULT ON NULL AS IDENTITY — позволяет указать значение. Если значение в столбце не указано или явно указать значение NULL, то сервер назначит значение из последовательности.



# Oracle Identity

```
INSERT INTO EMP(NAME, DEPT) VALUES ('King', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(1020, 'Edwards', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(NULL, 'Dow', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(NULL, 'Jones', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(20, 'Forrest', 20);
INSERT INTO EMP(NAME, DEPT) VALUES('Lucas', 20);
COMMIT;
```

	ID	NAME	DEPT
1	1000	King	20
2	1020	Edwards	20
3	1010	Dow	20
4	1020	Jones	20
5	20	Forrest	20
6	1030	Lucas	20



# Кластеры

---

- ▶ Таблицы, с которыми часто работают совместно, можно физически хранить совместно.
- ▶ Для этого создается кластер, который будет их содержать
- ▶ Строки из отдельных таблиц сохраняются в одних и тех же блоках, поэтому объединяющие запросы выполняются быстрее
- ▶ Уменьшается количество операций ввода-вывода
- ▶ Производительность операций вставки, обновления и удаления может быть ниже, чем для обычных таблиц
- ▶ Связанные столбцы называются кластерным ключом



# Кластеры

---

- ▶ Таблицы, с которыми часто работают совместно, можно физически хранить совместно.
- ▶ Для этого создается кластер, который будет их содержать
- ▶ Строки из отдельных таблиц сохраняются в одних и тех же блоках, поэтому объединяющие запросы выполняются быстрее
- ▶ Уменьшается количество операций ввода-вывода
- ▶ Производительность операций вставки, обновления и удаления может быть ниже, чем для обычных таблиц
- ▶ Связанные столбцы называются кластерным ключом



# Хэш-кластеры

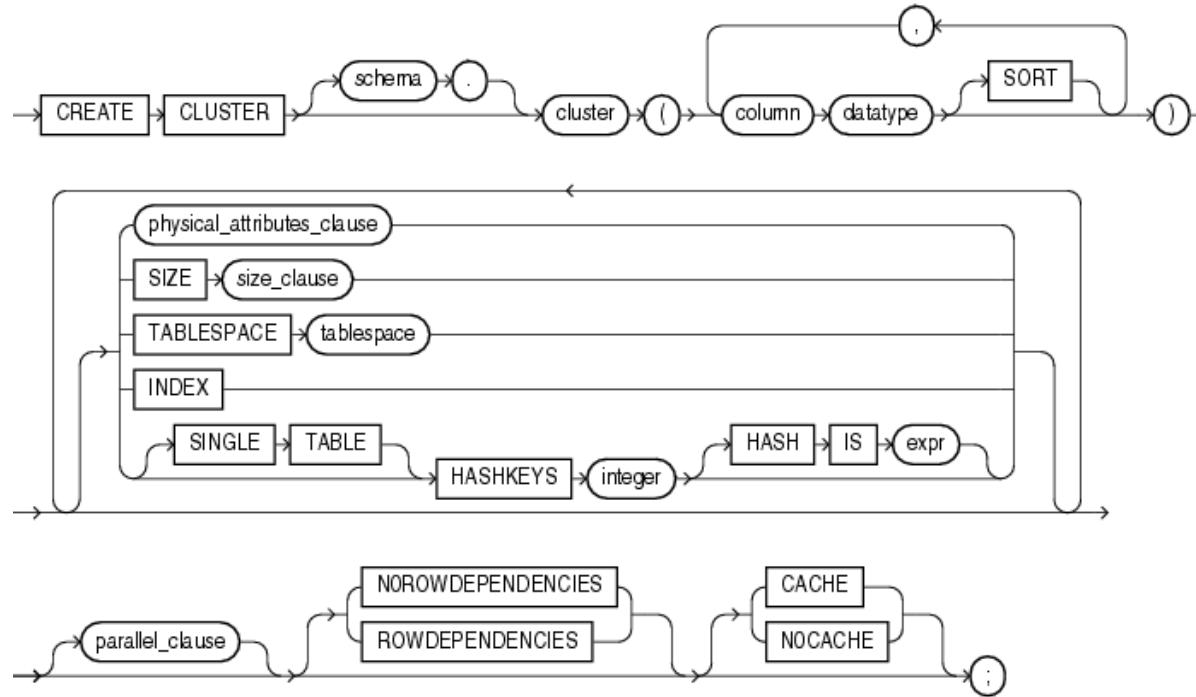
---

- ▶ Хэш-кластеры используют функции хэширования кластерного ключа строки для определения физической локализации места, где строку следует хранить
- ▶ Наибольшие преимущества – в запросах, использующих операции равенства:
- ▶ `select Name from STUDENT where Id = 999;`



# Кластеры

- ▶ Кластер – объект БД, который хранит значения общих столбцов нескольких таблиц
- ▶ Создание CREATE CLUSTER
- ▶ Привилегия CREATE CLUSTER



# Кластеры

```
SYSTEM@sh1> connect svvcore/svvcore@sh1;
Соединено.
sh1 - SVVCORE - 19.10.10
SVVCORE@sh1> create cluster svvcore.cluPULPIT
 2  (
 3    PULPIT CHAR(10)
 4  )
 5 hashkeys 100 ;
```

Кластер создан.

```
SVVCORE@sh1> CREATE TABLE SVVCORE.TEACHER
 2  (
 3    TEACHER      CHAR(10),
 4    TEACHER_NAME VARCHAR2(50),
 5    PULPIT       CHAR(10)
 6  )
 7 CLUSTER SVVCORE.cluPULPIT(PULPIT);
```

Таблица создана.

```
SVVCORE@sh1> CREATE TABLE SVVCORE.PULPIT
 2  (
 3    PULPIT      CHAR(10),
 4    PULPIT_NAME VARCHAR2(100),
 5    FACULTY     CHAR(10)
 6  )
 7 CLUSTER SVVCORE.cluPULPIT(PULPIT)
 8 COMPRESS
 9 -- CACHE не возможет с кластером
10 ;
```

 Таблица создана.

# Кластеры

```
SUVCORE@sh1> select CLUSTER_NAME,  
2      OWNER,  
3      TABLESPACE_NAME,  
4      CLUSTER_TYPE,  
5      CACHE  
6  from DBA_CLUSTERS;
```

CLUSTER_NAME	OWNER	TABLESPACE_NAME	CLUST	CACHE
C_COBJ#	SYS	SYSTEM	INDEX	N
C_TS#	SYS	SYSTEM	INDEX	N
C_FILE#_BLOCK#	SYS	SYSTEM	INDEX	N
C_USER#	SYS	SYSTEM	INDEX	N
C_OBJ#	SYS	SYSTEM	INDEX	N
C_MLOG#	SYS	SYSTEM	INDEX	N
C_TOID_VERSION#	SYS	SYSTEM	INDEX	N
C_RG#	SYS	SYSTEM	INDEX	N
C_OBJ#_INCOL#	SYS	SYSTEM	INDEX	N
SMON_SCN_TO_TIME	SYS	SYSTEM	INDEX	N
CLUPULPIT	SUVCORE	TSDDATASUU	HASH	N

11 строк выбрано.

# Кластеры

```
SUVCORE@sh1> select tablespace_name,
  2      segment_type,
  3      count(*),
  4      sum(blocks),
  5      sum(extents),
  6      sum(bytes)
  7 from dba_segments
  8 where tablespace_name='TSDATASUU' OR tablespace_name ='TSTEMPSUU'
  9 group by tablespace_name, segment_type
10 order by tablespace_name, segment_type;
```

TABLESPACE_NAME	SEGMENT_TYPE	COUNT(*)	SUM(BLOCKS)	SUM(EXTENTS)	SUM(BYTES)
TSDATASUU	<u>CLUSTER</u>	<u>1</u>	<u>104</u>	<u>13</u>	<u>851968</u>
TSDATASUU	<u>INDEX</u>	<u>10</u>	<u>80</u>	<u>10</u>	<u>655360</u>
TSDATASUU	TABLE	8	64	8	524288

# Синонимы

---

- ▶ Синоним – способ обращаться к объекту базы данных без указания обязательной полной идентификации объекта (хост – экземпляр – владелец – объект).
- ▶ Частный синоним принадлежит пользователю, который его создал.
- ▶ Публичный синоним используется совместно всеми пользователями базы данных.



# Синонимы

---

- ▶ Привилегия – CREATE (PUBLIC) SYNONYM
- ▶ Создание – CREATE (PUBLIC) SYNONYM
- ▶ Допустимость синонима не проверяется сервером при создании!
- ▶ Представление словаря dba.synonyms



# Синонимы

---

- ▶ Может указывать на:
  - ▶ Таблицы,
  - ▶ Процедуры,
  - ▶ Функции,
  - ▶ Последовательности,
  - ▶ Представления
  - ▶ Пакеты
  - ▶ Объекты в локальной или удаленной базе данных



# Синонимы

```
GRANT CREATE SYNONYM TO RLSVVCORE
```

```
CREATE SYNONYM T1 FOR SVVCORE.TEACHER;
```

```
SELECT * FROM T1;
```

```
DROP SYNONYM T1;
```

```
GRANT CREATE PUBLIC SYNONYM,
```

```
    DROP PUBLIC SYNONYM      TO RLSVVCORE
```

```
CREATE PUBLIC SYNONYM T1 FOR SVVCORE.TEACHER;
```

```
SQL> connect isuscore/isuscore@isus
```

```
Соединено.
```

```
SQL> select * from t1;
```



# Представления

---

- ▶ Представление – хранимый запрос
- ▶ Можно обращаться, как к обычной таблице
- ▶ Данные хранятся в таблице
- ▶ Добавляют уровень защиты данных
- ▶ Скрывают сложность данных
- ▶ Скрывают имена столбцов таблиц



# Представления

---

- ▶ Привилегия – CREATE VIEW
- ▶ Создание – CREATE (OR REPLACE) VIEW
- ▶ FORCE – создает представление, независимо от того, существуют ли таблицы и есть ли права
- ▶ NOFORCE – по умолчанию
- ▶ WITH CHECK OPTION – указывает, что будут вставлены или изменены строки, которые будут выбираться через это представление
- ▶ READ ONLY



# Представления

```
CREATE VIEW VPULPIT_NAME  
AS SELECT PUPIT_NAME FROM SVVCORE.PULPIT;
```

SQL Error: ORA-00942: таблица или представление пользователя не существует  
00942. 00000 - "table or view does not exist"

```
CREATE FORCE VIEW VPULPIT_NAME  
AS SELECT PUPIT_NAME FROM SVVCORE.PULPIT;
```

SQL Command: CREATE VIEW

Failed: Warning: выполнение завершено с предупреждением

```
SELECT * FROM VPULPIT_NAME;
```

ORA-04063: view "SVVCORE.VPULPIT\_NAME" имеет ошибки

```
SELECT OWNER, VIEW_NAME, TEXT FROM DBA_VIEWS WHERE OWNER='SVVCORE'
```

SVVCORE	VPULPIT_NAME	SELECT PUPIT_NAME FROM SVVCORE.PULPIT
---------	--------------	---------------------------------------

SVVCORE	VTEACHER_NAME	SELECT TEACHER_NAME FROM SVVCORE.TEACHER
---------	---------------	--



# Представления

```
CREATE TABLE SVVCORE.PULPIT  
(  
    PULPIT      CHAR(10) PRIMARY KEY,  
    PULPIT_NAME VARCHAR2(50)  
)  
  
SELECT * FROM VPULPIT_NAME;
```

ORA-04063: view "SVVCORE.VPULPIT\_NAME" имеет ошибки

```
ALTER VIEW vpulpit_name COMPILE
```

Failed: Warning: выполнение завершено с предупреждением  
ALTER VIEW vpulpit\_name succeeded.

```
CREATE OR REPLACE VIEW VPULPIT_NAME_FK  
AS SELECT PULPIT_NAME FROM SVVCORE.PULPIT WHERE PULPIT='ФК'  
WITH CHECK OPTION CONSTRAINT PULPIT_P;
```

# Материализованные представления

- ▶ Механизм переписывания запросов (QUERY REWRITE)
  - подмена запроса перед выполнением
  - ▶ *QUERY REWRITE ENABLED*
  - ▶ *QUERY REWRITE INTEGRITY*
    - ▶ *ENFORCED*
    - ▶ *TRUSTED*
    - ▶ *STALE TOLERATED*

```
SELECT NAME, VALUE FROM V$PARAMETER  
WHERE NAME LIKE '%rewrite%';
```

	NAME	VALUE
1	query_rewrite_enabled	TRUE
2	query_rewrite_integrity	enforced



# Материализованные представления

---

- ▶ Материализованное представление — физический объект базы данных, содержащий результат выполнения запроса
- ▶ Привилегия – CREATE MATERIALIZED VIEW
- ▶ Создание – CREATE MATERIALIZED VIEW
- ▶ BUILD IMMEDIATE – создает представление в момент выполнения оператора
- ▶ START WITH – показывает, когда выполнится в первый раз (если не был построен сразу)
- ▶ NEXT – показывает, когда выполнится в следующий раз
- ▶ Далее – в разницу времени между START WITH и NEXT



# Материализованные представления

---

```
GRANT CREATE MATERIALIZED VIEW TO A1;
```

```
GRANT QUERY REWRITE TO A1;
```

```
CREATE MATERIALIZED VIEW EMP_SUMMARY  
    BUILD IMMEDIATE  
    REFRESH COMPLETE  
    ENABLE QUERY REWRITE  
    AS SELECT COUNT(EMP_ID) FROM EMPLOYEE;
```



# Материализованные представления

---

- ▶ REFRESH
  - ▶ COMPLETE – полное обновление данных из базовых таблиц
  - ▶ REFRESH FAST – используются журналы фиксации изменений базовых таблиц
  - ▶ REFRESH FORCE – попытка быстрого обновления; если быстрое обновление невозможно, то выполняется полное обновление



# Материализованные представления

---

- ▶ Обновление явное
- ▶ Обновление неявное по расписанию
- REFRESH
  - ON COMMIT – обновление при COMMIT
  - ON DEMAND – обновление по требованию

```
ALTER MATERIALIZED VIEW EMP_SUMMARY
REFRESH COMPLETE ON DEMAND
NEXT SYSDATE + NUMTODSINTERVAL (2, 'MINUTE');

EXECUTE DEMS_MVIEW.REFRESH ('EMP_SUMMARY');
```



# Материализованные представления

---

- ▶ *REFRESH FAST*
- ▶ Создать журналы материализованного представления
  - вспомогательные таблицы, накапливающие сведения об изменениях, которые происходят в базовых таблицах
- ▶ Создать материализованное представление

```
CREATE MATERIALIZED VIEW LOG ON EMPLOYEE  
  WITH ROWID, PRIMARY KEY  
  INCLUDING NEW VALUES;
```

```
CREATE MATERIALIZED VIEW EMP_SALARY_LIST  
  REFRESH FAST ON COMMIT  
  ENABLE QUERY REWRITE  
  AS SELECT EMP_ID, SALARY FROM EMPLOYEE;
```



# Материализованные представления

---

- ▶ Просмотр
  - ▶ *USER\_MVIEWS*
  - ▶ *USER\_MVIEW\_LOGS*
- ▶ Удалить
  - ▶ *DROP MATERIALIZED VIEW*



# Индексы

---

- ▶ Индекс – структура базы данных, используемая сервером для быстрого поиска строки в таблице



# Оптимизация запросов

---

- ▶ Оптимизатор запроса — встроенное в СУБД программное обеспечение, которое определяет наиболее эффективный способ выполнения SQL-выражения
- ▶ План выполнения запроса — последовательность шагов или инструкций СУБД, необходимых для выполнения SQL-выражения
- ▶ Стоимость выполнения запроса — наилучшая оценка времени, необходимого для выполнения оператора, полученная оптимизатором



# Оптимизация запросов

---

- ▶ Оптимизатор разрабатывает множество потенциальных планов выполнения SQL-запросов
- ▶ Для каждого потенциального плана выполнения запроса оптимизатор оценивает стоимость его выполнения



# Оптимизация запросов

---

- ▶ Для оценки применяется информация о ресурсах:
  - ▶ дисковый ввод/вывод
  - ▶ загрузка центрального процессора
  - ▶ оперативная память
- ▶ Выбор оптимального плана выполнения запроса зависит от:
  - ▶ объема предполагаемого набора данных
  - ▶ расположение данных
  - ▶ структуры доступа к данным



# Оптимизация запросов

---

- ▶ Оптимизатор определяет наилучший план выполнения разными способами:
  - ▶ полное сканирование таблицы
  - ▶ использование индексов
  - ▶ различные типы соединения



# Оптимизация запросов

---

- ▶ Селективность (избирательность)
- ▶ Кардинальность



# Селективность

---

- ▶ Селективность таблицы — значение, представляющее долю строк таблицы, удовлетворяющих определенному условию выбора
- ▶ Селективность таблицы связана с условием выбора строк
- ▶ Селективность индекса — значение, представляющее отношение количества уникальных значений индексируемых столбцов к общему числу строк таблицы
- ▶ Селективность индекса показывает долю строк от общего числа строк в таблице, которое приходится на одно значение индекса



# Кардинальность

---

- ▶ Кардинальность — количество строк, возвращаемых после каждой операции плана выполнения запроса
- ▶ Значение кардинальности равно произведению селективности на количество обработанных строк



# План запроса

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's a code editor window with the following SQL query:

```
65  
66  
67 select ename, empno, deptno from emp  
68
```

Below the editor, there are three tabs: Script Output, Query Result, and Explain Plan. The Explain Plan tab is currently active.

On the right side of the screen, a context menu is open. It contains several options:

- Run Script F5
- Create Report...
- Save as Snippet...
- Autotrace F8
- Explain Plan... F10** (This option is highlighted with a red rectangle.)
- SQL Tuning Advisor... Ctrl-F12
- Commit F11
- Rollback F12
- To Upper/Lower/InitCap Ctrl+Quote
- Clear
- SQL History F8
- Cut Ctrl-X
- Copy Ctrl-C
- Paste Ctrl-V

# План запроса

```
67 | select ename, empno, deptno  
68 | from emp  
69 | where empno = 7839;  
70 |
```

Script Output × | Query Result × | Explain Plan ×

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
TABLE ACCESS	EMP	BY INDEX ROWID	1
INDEX	EMP_PK	UNIQUE SCAN	0
Access Predicates			
EMPNO=7839			

# План запроса

```
67 | select ename, empno, deptno  
68 | from emp  
69 | --where empno = 7839;  
70 |  
71 |
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
TABLE ACCESS	EMP	FULL	2

# План запроса

```
67 | select ename, empno, deptno  
68 | from emp  
69 | where deptno = 10;  
70 |  
71 |
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
TABLE ACCESS	EMP	FULL	2
Filter Predicates			
DEPTNO=10			

# План запроса

```
304 select c.company, o.order_num, o.order_date, o.amount  
305 from customers c join orders o on c.cust_num = o.cust  
306 where o.amount <50;
```

307

Script Output x | Query Result x | Explain Plan x

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			3
HASH JOIN			3
Access Predicates			
C.CUST_NUM=O.CUST			
NESTED LOOPS			3
NESTED LOOPS			3
STATISTICS COLLECTOR			
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
O.AMOUNT<50			
INDEX	SYS_C0010526	UNIQUE SCAN	0
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1
TABLE ACCESS	CUSTOMERS	FULL	1

# Индексы

---

- ▶ Типы индексов:
  - ▶ Табличный (B\*Tree) индекс
  - ▶ Битовый индекс
  - ▶ Функциональный индекс



# Индексы

---

- ▶ Табличный индекс (B\*Tree) структурирован в виде сбалансированного дерева
- ▶ Листовой блок содержит индексированные значения столбца и соответствующий ему идентификатор строки (RowId)
- ▶ Предназначен для индексирования уникальных столбцов или столбцов с высокой селективностью

```
| create index or_amount  
| on orders(amount);
```



# Индексы

---

- ▶ Битовый индекс создает битовые карты для каждого возможного значения столбца, где каждому биту соответствует строка, а значение бита 1 (0) означает, что соответствующая строка содержит (не содержит) индексируемое значение
- ▶ Предназначен для индексирования столбцов с низкой селективностью
- ▶ Не подходит для таблиц с частым обновлением
- ▶ Хорошо подходят для хранилищ данных



# Индексы

```
SUVCORE@sh1> create bitmap index ibmAUDITORIUM_TYPE on AUDITORIUM(AUDITORIUM_TYPE);
```

Индекс создан.

```
SUVCORE@sh1> create index idxTEACHER_NAME on TEACHER(TEACHER_NAME ASC);
```

Индекс создан.

```
SUVCORE@sh1> select index_name,  
2      index_type,  
3      table_name,  
4      uniqueness  
5  from DBA_INDEXES  
6  where owner = 'SUVCORE'  
7  /
```

INDEX_NAME	INDEX_TYPE	TABLE_NAME	UNIQUENESS
?K_FACULTY	TOT - TOP	FACULTY	UNIQUE
?K_PULPIT	NORMAL	PULPIT	UNIQUE
idxTEACHER_NAME	NORMAL	TEACHER	NONUNIQUE
?K_TEACHER	NORMAL	TEACHER	UNIQUE
?K_AUDITORIUM	NORMAL	AUDITORIUM	UNIQUE
ibmAUDITORIUM_TYPE	BITMAP	AUDITORIUM	NONUNIQUE
?K_AUDITORIUM_TYPE	NORMAL	AUDITORIUM_TYPE	UNIQUE
?K SUBJECT	NORMAL	SUBJECT	UNIQUE

3 строк выбрано.

# Индексы

---

- ▶ Функциональный индекс – предварительно вычисляют значения функции по заданному столбцу и сохраняют результат в индексе
- ▶ LOWER(NAME) / UPPER (NAME)



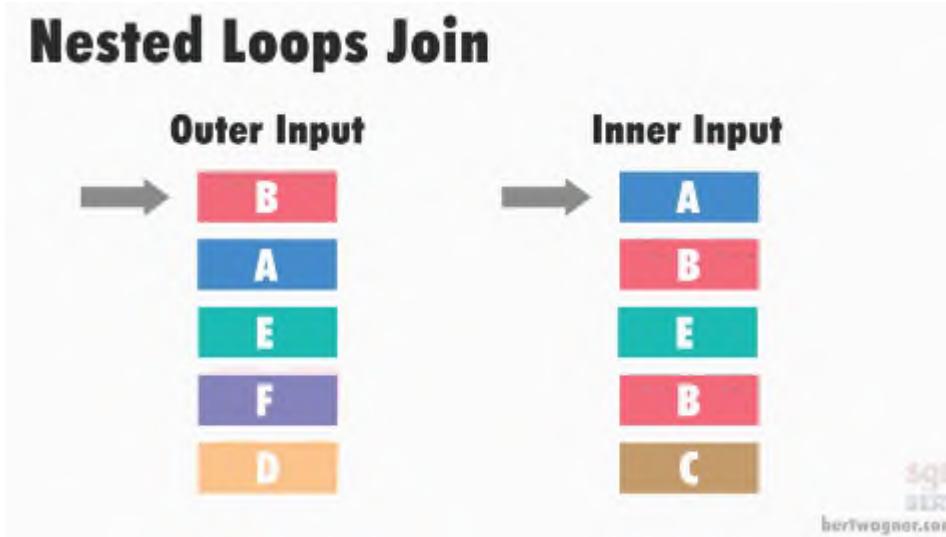
# Индексы

---

- ▶ Функциональный индекс – предварительно вычисляют значения функции по заданному столбцу и сохраняют результат в индексе
- ▶ LOWER(NAME) / UPPER (NAME)

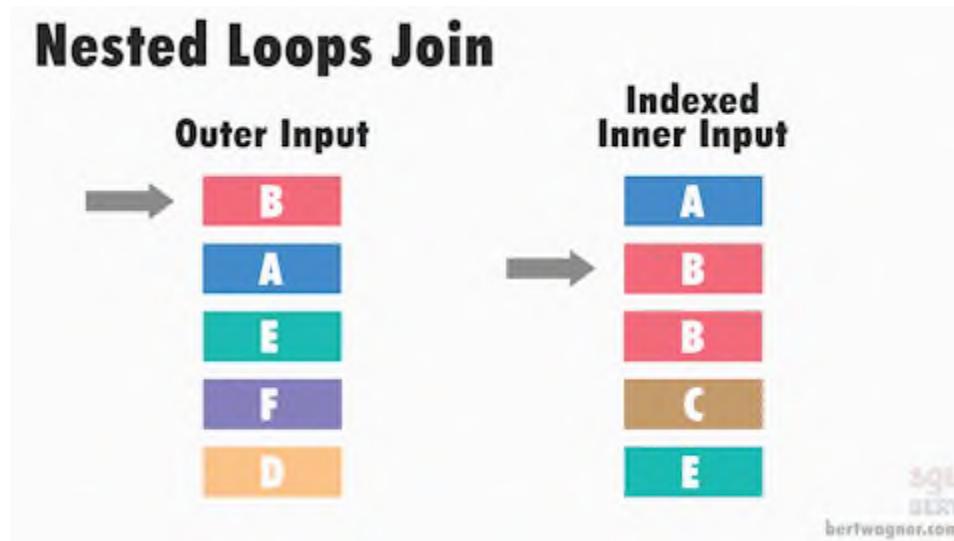


# Nested Loops Join

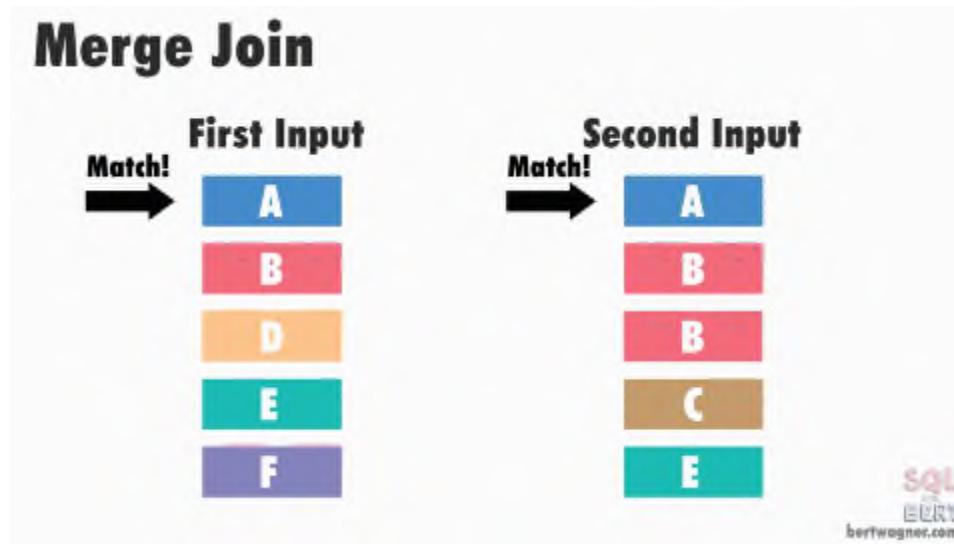


<https://bertwagner.com/>

# Indexed Nested Loops Join

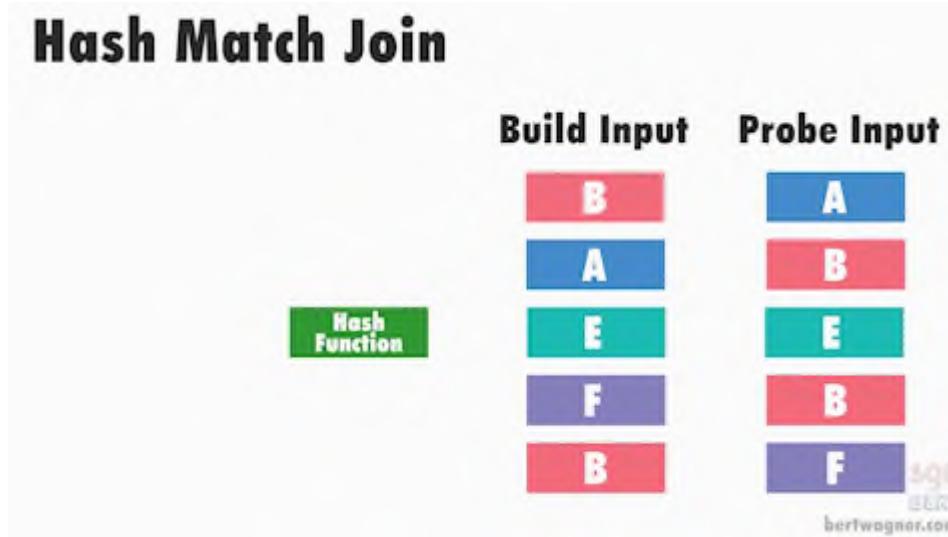


# Merge Join



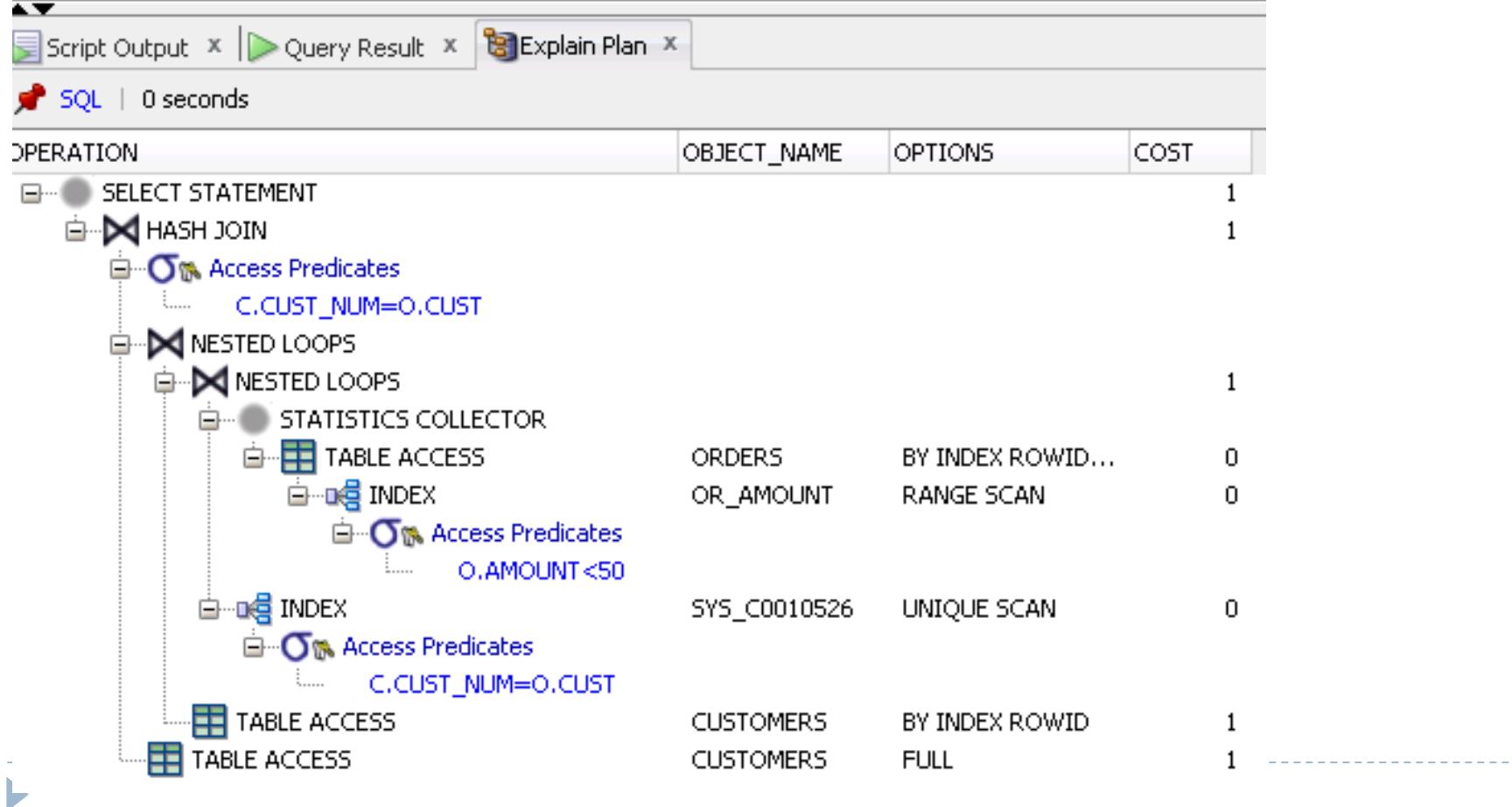
# Hash Join

## Hash Match Join



# План запроса

```
304 | select c.company, o.order_num, o.order_date, o.amount  
305 | from customers c join orders o on c.cust_num = o.cust  
306 | where o.amount <50;  
307 |
```



# План запроса

```
304 | select c.company, o.order_num, o.order_date, o.amount  
305 | from customers c join orders o on c.cust_num = o.cust  
306 | where o.amount >50;
```

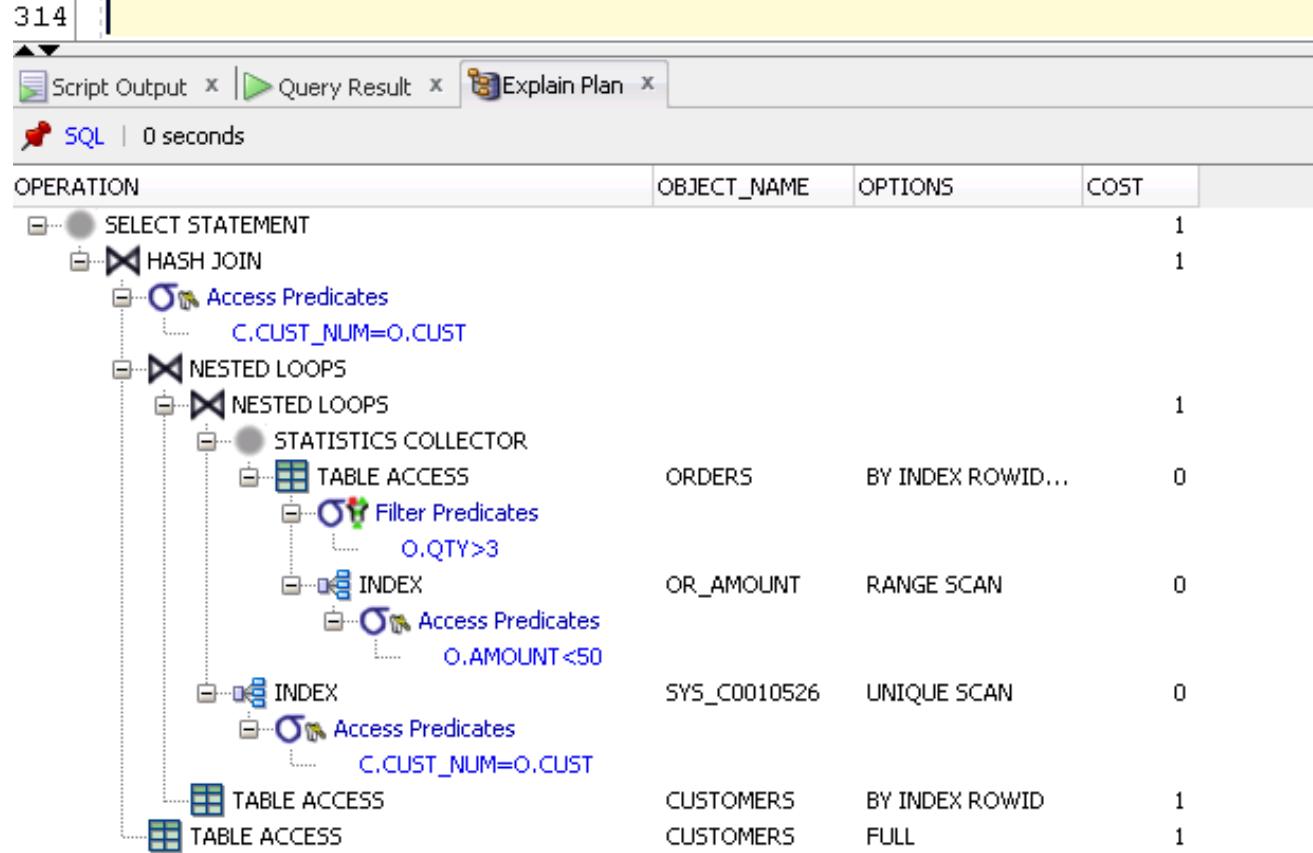
Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			4
HASH JOIN			4
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	FULL	2
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
O.AMOUNT>50			

# План запроса

```
311 | select c.company, o.order_num, o.order_date, o.amount  
312 | from customers c join orders o on c.cust_num = o.cust  
313 | where o.amount < 50 and o.qty >3;  
314 |
```



# План запроса

```
311 | select c.company, o.order_num, o.order_date, o.amount  
312 | from customers c join orders o on c.cust_num = o.cust  
313 | where o.amount < 50 or o.qty >3;  
314 |
```

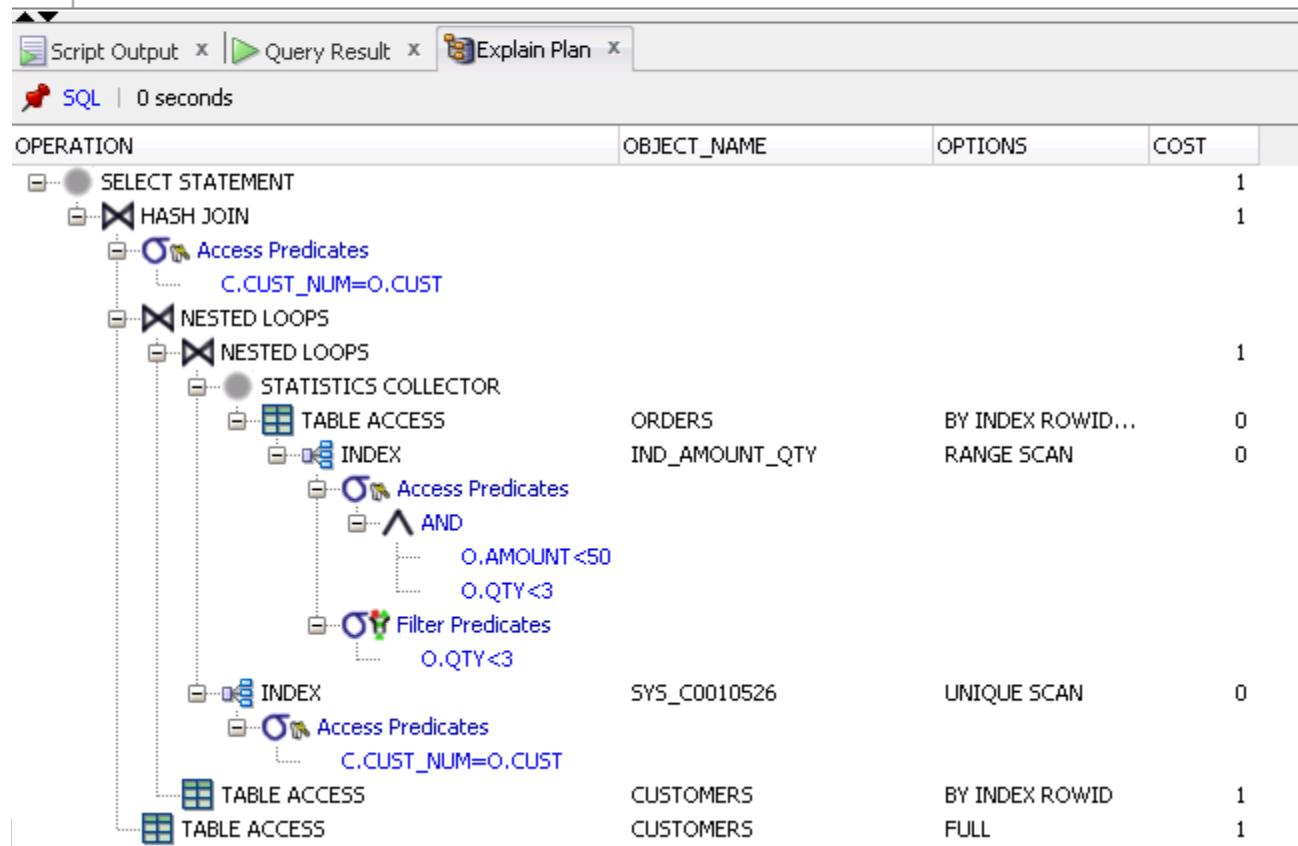
Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			4
HASH JOIN			4
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	FULL	2
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
OR			
O.AMOUNT<50			
O.QTY>3			

# План запроса

```
311  create index ind_amount_qty  
312  on orders(amount, qty);  
313  
314  select c.company, o.order_num, o.order_date, o.amount  
315  from customers c join orders o on c.cust_num = o.cust  
316  where o.amount < 50 and o.qty <3;  
317
```



# Database Link

---

- ▶ Database Link (dblink) - объект базы данных, предназначенный для доступа к объектам базы данных, управляемой другим сервером
- ▶ Чтобы иметь возможность создать dblink, необходимо выдать привилегии:
- ▶ GRANT CREATE DATABASE LINK TO **USERNAME**



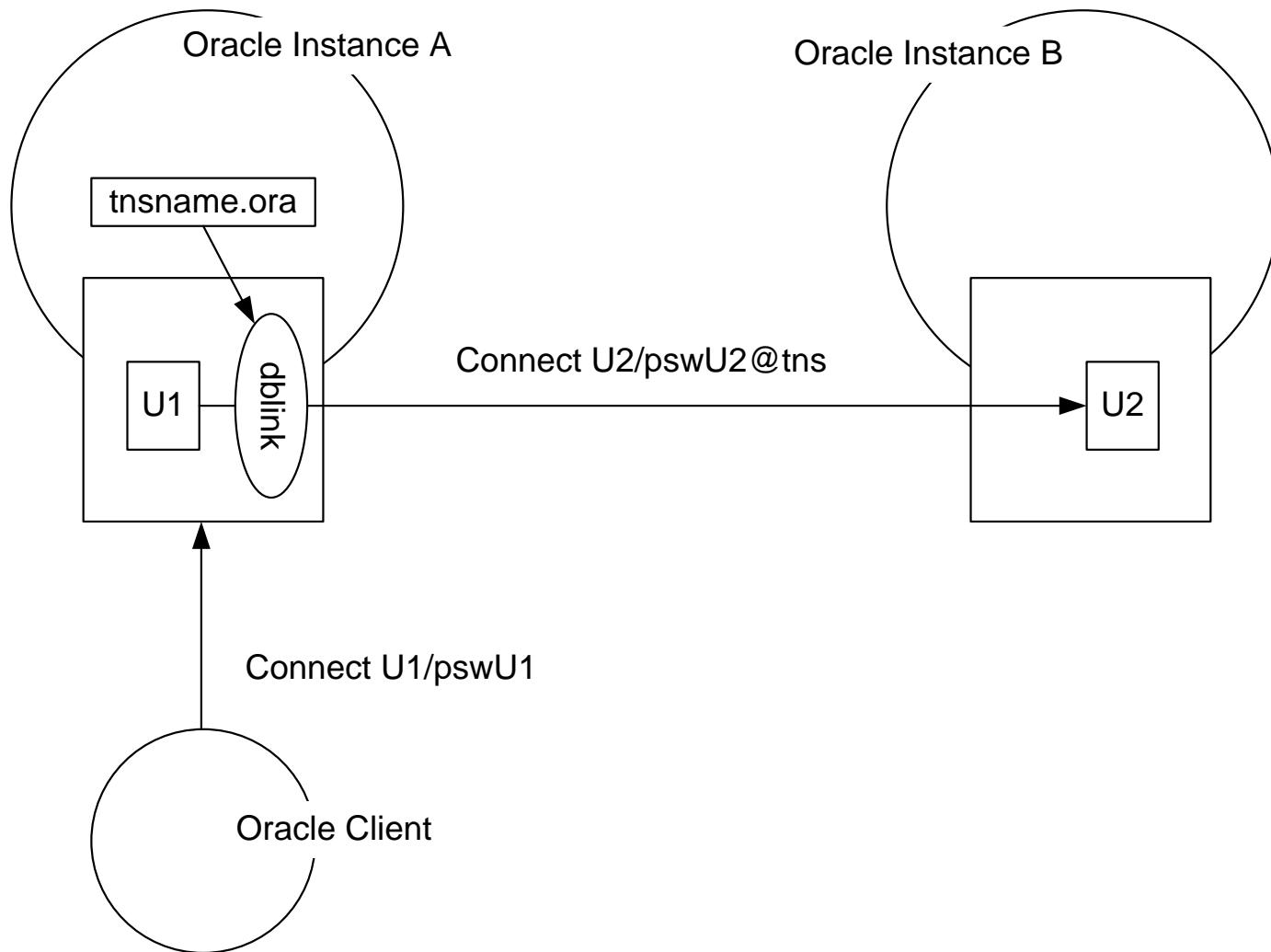
# Database Link

---

- ▶ Чтобы создать dblink типа user1-user2:
- ▶ **CREATE DATABASE LINK anotherdb**
- ▶ **CONNECT TO **USER2****
- ▶ **IDENTIFIED BY **PASSWORD****
- ▶ **USING '**INST\_B**';**
- ▶ '**INST\_B**' - сетевое имя для удаленной БД средствами Oracle Net



# Database Link

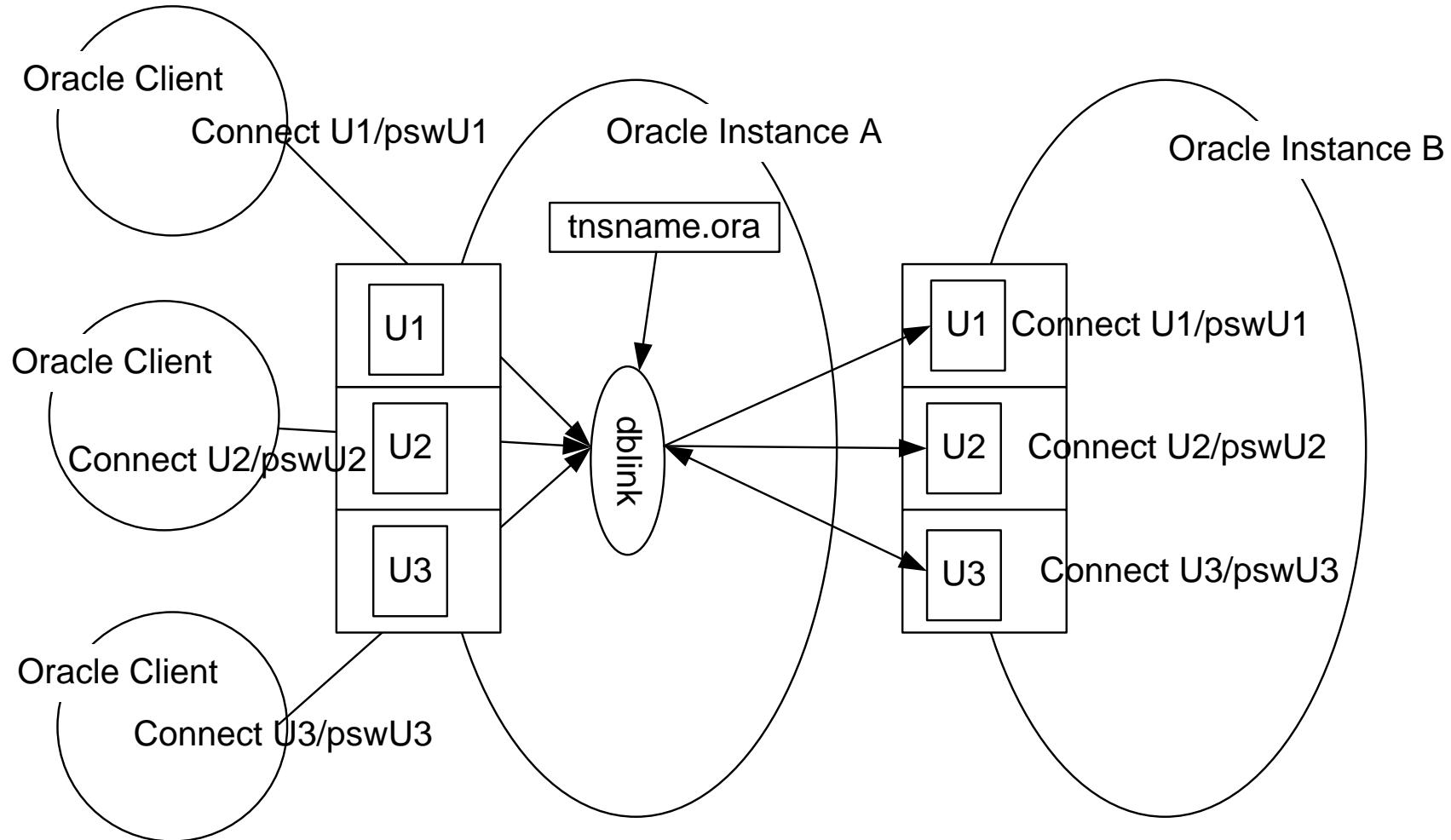


# PUBLIC DATABASE LINK

---

- ▶ Чтобы иметь возможность создать dblink типа global, необходимо выдать привилегии:
  - ▶ GRANT CREATE PUBLIC DATABASE LINK,
  - ▶ DROP PUBLIC DATABASE LINK TO **USERNAME**
  - ▶
  - ▶ Чтобы создать dblink типа global:
  - ▶ CREATE PUBLIC DATABASE LINK **public\_anotherdb**
  - ▶ **USING 'INST\_B';**
-

# PUBLIC DATABASE LINK



# DATABASE LINK

---

- ▶ Чтобы обратиться к объектам удаленного сервера, необходимо:
- ▶ **SELECT name FROM table\_name@anotherdb;**



# Database Link

---

- ▶ Чтобы закрыть dblink (открыт на время сессии):
- ▶ **ALTER SESSION CLOSE DATABASE LINK *anotherdb*;**



# Database Link

---

- ▶ Словарь для dblink:
- ▶ DBA\_DB\_LINKS
- ▶ USER\_DB\_LINKS
- ▶ V\$DBLINKS



# Database Link

---

- ▶ Параметры Oracle для dblink v\$PARAMETER:
- ▶ open\_links – максимальное кол-во открытых соединений (dblink) в одной сессии.
- ▶ open\_links\_per\_instance – максимальное количество соединений для одного экземпляра Oracle.



# Вопросы?

---



# **Архитектура ORACLE**

**Управление безопасностью**

**Лекция 9**

# Привилегии и роли

---

- ▶ GRANT / REVOKE
- ▶ Системные и объектные привилегии
- ▶ Нельзя выдавать в одном предложении



# Системные привилегии

---

- ▶ WITH ADMIN OPTION – дают право пользователю также назначать/отбирать привилегии
  - ▶ ALTER
  - ▶ ANALYZE
  - ▶ AUDIT
  - ▶ BACKUP
  - ▶ CREATE
  - ▶ DROP
  - ▶ SELECT
- ▶ ANY – для любого объекта
- ▶ ALL – для всех объектов



# Объекты грантов для системных привилегий

---

- ▶ DATABASE
- ▶ USER
- ▶ PROFILE
- ▶ TABLESPACE
- ▶ ROLE
- ▶ TABLE
- ▶ INDEX
- ▶ TRIGGER
- ▶ PROCEDURE
- ▶ SEQUENCE
- ▶ VIEW



# Объектные привилегии

---

- ▶ WITH GRANT OPTION – дают право пользователю также назначать/отбирать привилегии
  - ▶ ALTER
  - ▶ DELETE
  - ▶ EXECUTE
  - ▶ INSERT
  - ▶ UPDATE
  - ▶ SELECT
  - ▶ REFERENCES
- ▶ Снимает привилегии тот, кто их назначил



# Объекты грантов для объектных привилегий

---

- ▶ TABLE
- ▶ VIEW
- ▶ SEQUENCE
- ▶ PROCEDURE



# Аудит

---

- ▶ Определение неавторизованного доступа
- ▶ Должен быть постоянно активен для:
  - ▶ Проверки доступа пользователей (особенно неуспешные попытки и доступ в нерабочее время)
  - ▶ Изменений структуры базы данных (пользователи не могут изменять структуру БД, администратор должен вносить изменения в специально отведенное время)
  - ▶ Использования системных привилегий
- ▶ Для успешных (*successful*) и для неуспешных (*not successful*) попыток
- ▶ Необходимо включать только для необходимых объектов
- ▶ По умолчанию выключен



# Уровни аудита

---

- ▶ На уровне объектов схемы
  - ▶ На уровне SQL- выражений
  - ▶ На уровне системных привилегий
- 
- ▶ Дополнительно:
    - ▶ BY username – будут фиксироваться действия только конкретного пользователя
    - ▶ WHENEVER SUCCESSFUL / WHENEVER NOT SUCCESSFUL – фиксируются успешные или безуспешные попытки
    - ▶ BY SESSION / BY ACCESS – как фиксируются действия: одна запись на сеанс или одна на каждое действие в течение сеанса



# Детальный аудит

---

- ▶ Системные триггеры
- ▶ DML-триггеры
- ▶ Детализированный аудит (fine-grained) – пакет dbms\_fga



# Конфигурация аудита

---

- ▶ Включение аудита в параметрах: `audit_trail = db`
- ▶ Проверка, что аудит включен:  
`select name, value from v$parameter where name like 'audit%';`
- ▶ Пока ничего не отслеживается, кроме важных действий  
(подключения привилегированных пользователей, запуск, останов,  
добавление файла данных)
- ▶ Изменения в `oracle_home/rdbms/audit` по умолчанию
- ▶ Можно переопределить `audit_file_dest` в файле параметров



# Проверка отслеживаемых действий аудита

---

- ▶ Проверка, что выражения отслеживаются:

```
select * from dba_stmt_audit_opts
```

- ▶ Проверка, что привилегии отслеживаются:

```
select * from dba_priv_audit_opts;
```

- ▶ Проверка, что объекты отслеживаются:

```
select * from dba_obj_audit_opts;
```



# Формат команды включения аудита

---

- ▶ `audit { statement_option | privilege_option } [ by user ]  
[ by {session | access} ] [whenever { successful | un successful } ]`
  
- ▶ Аудит могут задать пользователи с привилегией audit system:  
`select * from dba_sys_privs where privilege like '%audit%';`
  
- ▶ Примеры:
- ▶ `audit create session;`
- ▶ `audit create table;`
- ▶ `audit create index;`
- ▶ `audit alter trigger;`
- ▶ `audit drop table;`
- ▶ `audit execute procedure;`



# Контроль журнала аудита

---

- ▶ Каждая запись аудита добавляет запись в системную таблицу aud\$
  - табличное пространство system, владелец – sys
- ▶ Только sys может удалять данные из aud\$
- ▶ Роли select\_catalog\_role, delete\_catalog\_role имеют право доступа к aud\$ (добавить пользователя)
- ▶ Возможная атака – отказ в обслуживании (из-за роста размера таблицы и ограничения размера табличного пространства system)
- ▶ Тактика – регулярно копировать данные из aud\$ и усекать таблицу
- ▶ Выборки:
  - ▶ select \* from sys.aud\$;
  - ▶ select \* from dba\_audit\_trail;
  - ▶ select \* from dba\_audit\_session;



# Контроль журнала аудита

---

- ▶ Неудачные попытки входа (количество):

- ▶ 

```
select count(*) username, terminal, to_char(timestamp,'dd/mm/yyyy')
  from dba_audit_session
 where returncode <>0
 group by username, terminal, to_char(timestamp,'dd/mm/yyyy');
```

- ▶ Все попытки входа (количество):

- ▶ 

```
select count(*) username, terminal, to_char(timestamp,'dd/mm/yyyy')
  from dba_audit_session
 group by username, terminal, to_char(timestamp,'dd/mm/yyyy');
```

- ▶ Попытки входа несуществующих пользователей:

- ▶ 

```
select username, terminal, to_char(timestamp,'dd/mm/yyyy')
  from dba_audit_session
 where returncode <>0 and
       not exist (select 1 from dba_users
                  where dba_users.username = dba_audit_session.username);
```



# Контроль журнала аудита

---

- ▶ Попытки входа в нерабочее время:
  - ▶ 

```
select username, terminal, returncode,
       to_char(timestamp, 'DD-MON-YYYY HH24:MI:SS'),
       to_char(logoff_time, 'DD-MON-YYYY HH24:MI:SS')
  from dba_audit_session
 where timestamp between to_date('08:00:00','HH24:MI:SS') and
       to_date('19:00:00','HH24:MI:SS');
```
- ▶ Попытки входа с разных компьютеров:
  - ▶ 

```
select count(distinct (terminal)), username from dba_audit_session
 having count(distinct (terminal)) > 1
 group by username;
```
- ▶ Попытки входа разных пользователей с одного терминала:
  - ▶ 

```
select count(distinct (username)), terminal from dba_audit_session
 having count(distinct (username)) > 1
 group by terminal;
```



# Контроль журнала аудита

---

- ▶ Контроль изменения объектов (работа программистов!):

```
▶ select username,
       priv_used,
       obj_name,
       to_char(timestamp, 'DD-MON-YYYY HH24:MI:SS'),
       returncode
  from dba_audit_trail
 where priv_used is not null and
       priv_used <> 'CREATE SESSION';
```

# Аудит – итоги

---

- ▶ Аудит – часть общего плана обеспечения безопасности БД
- ▶ Не предоставлять привилегий в производственной БД обычным пользователям, только в рамках согласованных ролей, обычно `execute procedure`
- ▶ Удалить, заблокировать, изменить пароли всех учеток по умолчанию (типичные атаки)
- ▶ Расчистить роль `public` – убрать большинство привилегий
- ▶ План по обслуживанию журнала аудита – перенос данных, очистка, контроль ежедневно
- ▶ Детальный аудит – `fga` и триггеры



# Привилегии и роли

---

- ▶ CASCADE – каскадный отзыв
  
- ▶ REVOKE REFERENCES ON dept
- ▶ FROM skott
- ▶ CASCADE CONSTRAINTS;



# Вопросы?

---



# **ORACLE 12c**

**PL/SQL**

**Лекция 10**

# Характеристика языка PL/SQL

---

- ▶ Procedural Language extensions to SQL;
- ▶ Основной язык для программирования хранимых процедур (*stored procedures*);
- ▶ Интегрирован с базой данных Oracle;
- ▶ Производительность серверных модулей;
- ▶ Приложение может быть проще в реализации при написании бизнес-логики на основе хранимых процедур;
- ▶ Отсутствие накладных расходов на приведение типов;
- ▶ Может выполняться независимо от пользователя;
- ▶ PL/SQL-функции можно вызывать из SELECT запросов



# Характеристика языка PL/SQL

---

- ▶ Взаимодействие с пользователем (user interaction);
- ▶ Внутренний язык (proprietary for Oracle);
- ▶ Содержит элементы объектно-ориентированного программирования;
- ▶ Позволяет использовать объектные типы;
- ▶ Интерпретация (режим по умолчанию);
- ▶ Компиляция (промежуточный код на C и конечный объектный код процессора);
- ▶ Среда выполнения: SQL\*Plus, SQL Developer, TOAD.



# DBMS\_output

---

- ▶ Модуль DBMS\_OUTPUT обеспечивает вывод информации для отладки
- ▶ Владелец – пользователь SYS.
- ▶ Принципы работы модуля DBMS\_OUTPUT следующий:
  - ▶ Операция PUT берет свои аргументы и помещает во внутренний буфер для хранения.
  - ▶ Операция GET считывает этот буфер и возвращает его содержимое процедуре в качестве аргумента.
  - ▶ Размер буфера устанавливается с помощью процедуры ENABLE.
- ▶ DBMS\_OUTPUT.put\_line();



# DBMS\_output

Oracle SQL Developer : D:\sql\Oracle\10\xx.sql

File Edit View Navigate Run Versioning Tools Help

Connections x Re... x

xx.sql x

SQL Worksheet History

shl-svvcore

Connections  
shl  
shl-svvcore  
Tables (Filtered)  
Views  
Indexes  
Packages  
Procedures  
Functions  
Queues  
Queues Tables  
Triggers  
Types  
Sequences  
Materialized Views  
Materialized Views Logs  
Synonyms  
Public Synonyms  
Database Links  
Public Database Links  
Directories  
Java  
XML Schemas  
XML DB Repository  
Recycle Bin  
Jobs  
Other Users

Versioning Navigator x

Subversion

-- Oxx.sql

declare

n1 number(20,11) := 123456789.12345678911;  
n2 number(20,10) := 123456789.12345678911;  
n3 number(20) := 123456789.12345678911;  
n4 number(20,-1) := 123456789.12345678911;  
n5 number(20,-2) := 123456789.12345678911;  
n6 number(20,-21) := 123456789.12345678911;  
n7 number(20,-25) := 123456789.12345678911;  
n8 number(20,-83) := 123456789.12345678911;  
n9 number(20,-7) := 0.12345678911;  
n10 number(38,10) := 123456789.12345678911;  
n11 number(38,10) := 123456789E-12;

begin

dbms\_output.put\_line('n1 ='||n1);  
dbms\_output.put\_line('n2 ='||n2);  
dbms\_output.put\_line('n3 ='||n3);  
dbms\_output.put\_line('n4 ='||n4);  
dbms\_output.put\_line('n5 ='||n5);  
dbms\_output.put\_line('n6 ='||n6);  
dbms\_output.put\_line('n7 ='||n7);

Dbms Output x  
Buffer Size: 20000  
Poll

shl-svvcore x

Script Output x  
Task completed in 0,031 seconds

anonymous block completed  
anonymous block completed

Log

Saved: D:\sql\Oracle\10\xx.sql

Line 14 Column 22 | Insert | Windows: CR/LF Editing

n1 =123456789,12345678911  
n2 =123456789,12345678911  
n3 =123456789  
n4 =123456790  
n5 =123456800  
n6 =0  
n7 =0  
n8 =0  
n9 =0  
n10 =123456789,1234567891  
n11 =,0001234568

# Схема блока PL/SQL

Заголовок

IS

Раздел объявлений

BEGIN

Исполняемый раздел

EXCEPTION

Раздел исключений

END;



# Схема блока PL/SQL

```
PROCEDURE get_happy (ename_in IN VARCHAR2) •— Заголовок
IS
    l_hiredate DATE; •— Раздел объявлений
BEGIN
    l_hiredate := SYSDATE - 2;
    INSERT INTO employee
        (emp_name, hiredate)
    VALUES (ename_in, l_hiredate); •— Исполняемый раздел
EXCEPTION
    WHEN DUP_VAL_IN_INDEX
    THEN
        DBMS_OUTPUT.PUT_LINE
            ('Cannot insert.');?>
END;
```

►

# Анонимный блок PL/SQL

---

- ▶ Не имеет секции заголовка
- ▶ Используется как скрипт для выполнения PL/SQL выражений
- ▶ Не может быть вызван из другого блока
- ▶ Начинается с **DECLARE** или **BEGIN**
- ▶ Варианты использования:
  - ▶ Триггер на стороне клиента (**Oracle Development Tools**)
  - ▶ Триггер базы данных (содержит АБ)
  - ▶ SQL-скрипт (описание процедур, функций и **execute**)
  - ▶ Откомпилированная программа (блок в **execute** команде, выполняющейся на сервере)



# Анонимный блок PL/SQL

---

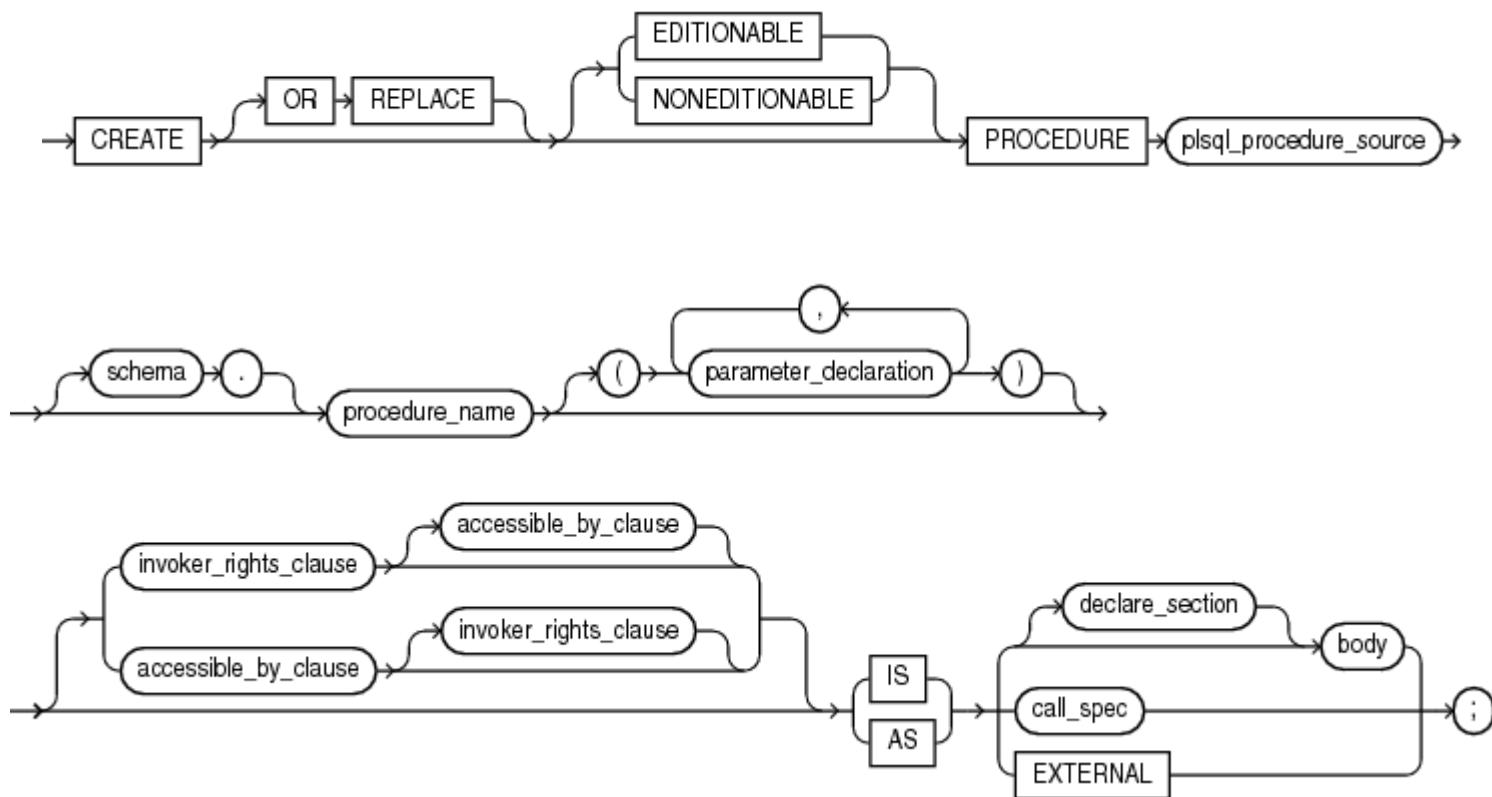
```
-- 01.sql
-- простейший анонимный блок PL/SQL
begin -- начало блока
    -- начало секции выполняемых команд (EXECUTION section)
null; --
    -- конец секции выполняемых команд
end; -- конец блока
/
```

```
-- 02.sql
-- PL/SQL не чувствителен к регистру
-- простейший анонимный блок PL/SQL
BEGIN -- начало блока
    -- начало секции выполняемых команд (EXECUTION section)
null; --
    -- конец секции выполняемых команд
end; -- конец блока
/
```



# Именованные блоки PL/SQL

- ▶ Процедуры
- ▶ Функции



# Секция объявлений

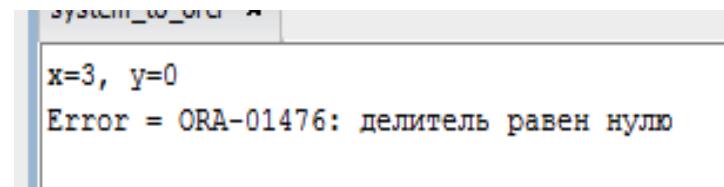
---

```
-- 03.sql
declare          -- необязательная секция объявления
    x number(3) := 3;  -- объявление переменной (область видимости блок)
begin
    dbms_output.put_line('x = '||x);  -- стандартный пакет процедур dbms_output
                                    -- процедура put_line - вывод ( serveroutput on)
                                    -- || конкатенация + неявное преобразование
end;
/
```



# Секция исключительных ситуаций

```
-- 04.sql
declare
    x number(3) := 3;
    y number(3) := 0;
    z number(10,2);
begin
    dbms_output.put_line(
        'x = ' || x ||
        ', y = ' || y
    );
    z := x/y;                      -- деление на 0
    dbms_output.put_line('z = ' || z); -- не выполняется
exception
    when others                  -- секция исключительных ситуаций
        then dbms_output.put_line('error = ' || sqlerrm);
                -- sqlerrm - зарезервированное имя переменной окружения
end;
/
```



```
x=3, y=0
Error = ORA-01476: делитель равен нулю
```

## Sqlerrm и sqlcode

---

- ▶ Функция SQLERRM возвращает сообщение об ошибке, связанной с исключительной ситуацией
- ▶ Функция SQLCODE возвращает номер ошибки, связанной с исключительной ситуацией
- ▶ Могут быть использована только в разделе обработки исключений
- ▶ Не имеют параметров или аргументов



# Секция исключительных ситуаций

---

- ▶ Может содержать столько блоков WHEN, сколько выделяется обрабатываемых исключений
- ▶ Остальные – в WHEN OTHERS
- ▶ Можно определять свои исключения

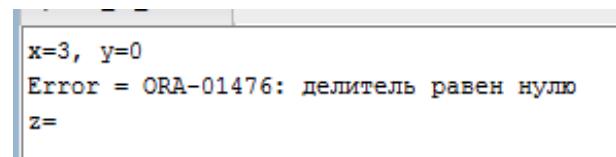
```
EXCEPTION
    WHEN exception_name
    THEN
        операторы обработки ошибки;
    . . .
    [WHEN OTHERS
    THEN
        обработка исключения по умолчанию; ]
```



# Вложенные блоки

- ▶ Область действия (scope) – переменные, исключения, модули – локальны в рамках блока;
- ▶ Область видимости – в текущем блоке;

```
-- 05.sql
declare
    x number(3) := 3;
    y number(3) := 0;
    z number(10,2);
begin
    dbms_output.put_line('x = '||x||', y = '||y );
    begin
        z := x/y;
    exception
        when others
            then dbms_output.put_line('error = '||sqlerrm);
    end;
    dbms_output.put_line('z = '||z);
end;
/
```



```
x=3, y=0
Error = ORA-01476: делитель равен нулю
z=
```

# Предупреждения компилятора

---

- ▶ ALL (все);
- ▶ PERFORMANCE (производительность);
- ▶ INFORMATIONAL (информационные);
- ▶ SEVERE (логика программы);
- ▶ Specific error (ошибка);



# Предупреждения компилятора

```
show parameter plsql_warnings;
```

Script Output x Query Result x  
Task completed in 0,046 seconds

NAME	TYPE	VALUE
plsql_warnings	string	DISABLE:ALL

```
select name, value from v$parameter where name = 'plsql_warnings';
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0,016 seconds

NAME	VALUE
1 plsql_warnings	DISABLE:ALL

# Предупреждения компилятора

```
alter system set plsql_warnings = 'ENABLE:INFORMATIONAL';

select name, value from v$parameter where name = 'plsql_warnings';
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0,016 seconds

NAME	VALUE
1 plsql_warnings	ENABLE:INFORMATIONAL, DISABLE:PERFORMANCE, DISABLE:SEVERE

```
SELECT DBMS_WARNING.GET_WARNING_SETTING_STRING FROM DUAL;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0 seconds

GET_WARNING_SETTING_STRING
1 ENABLE:INFORMATIONAL,DISABLE:PERFORMANCE,DISABLE:SEVERE

# Идентификаторы

---

- ▶ Идентификатор – наименование объекта PL/SQL
  - ▶ Константы
  - ▶ Скалярные переменные
  - ▶ Составные переменные:
    - ▶ Структуры
    - ▶ Коллекции
  - ▶ Исключения
  - ▶ Пакеты, процедуры и функции
  - ▶ Типы
  - ▶ Курсоры
  - ▶ Зарезервированные слова
  - ▶ Метки



# Идентификаторы

---

- ▶ Не более 30 символов
- ▶ Начинается с буквы
- ▶ Не содержит пробелов
- ▶ Может включать \$ \_ #
- ▶ Компилятор приводит идентификаторы к верхнему регистру
- ▶ “идентификатор” регистрозависим



# Спецсимволы и зарезервированные слова PL/SQL

```
1 select keyword from v$reserved_words
2* where length = 1 and keyword != 'A'
3 /
```

KEYWORD	KEYWORD	KEYWORD
+	=	*
<	.	)
^	&	(
P	T	E
,	U	
M	-	
/	[	
!	—	
:	>	
G	]	
@	K	



# Спецсимволы и зарезервированные слова PL/SOL

```
1 select keyword from v_$reserved_words
2* where length > 1 and keyword != 'A' order by keyword
SYS@sh1> /
```

KEYWORD

```
--<<
ABORT
ACCESS
ACCESSED
ACCOUNT
ACTIVATE
ADD
ADMIN
ADMINISTER
ADMINISTRATOR
ADVISE
```

KEYWORD

```
--ADVISOR
AFTER
ALIAS
ALL
ALLOCATE
ALLOW
ALL_ROWS
ALTER
ALWAYS
ANALYZE
ANCTILARY
```

```
WRAPPED
WRITE
$_DYN_PRUNE
$_ID
$_MLATTRIBUTES
$_MLCOLATTVAL
$_MLELEMENT
$_MLFOREST
$_MLNAMESPACES
```

KEYWORD

```
$_MLPARSE
$_MLPI
$_MLQUERY
$_MLROOT
$_MLSCHEMA
$_MLSERIALIZE
$_MLTABLE
$_MLTYPE
$_YEAR
$_YES
$_ZONE
```

1100 строк выбрано.

1000-1010



# Литералы

---

- ▶ Литерал – значение идентификатора
  - ▶ Number – 123, 21.6, NULL
  - ▶ String – ‘sentence’, ’01-01-2017’ , NULL
  - ▶ Boolean – true, false, NULL
  - ▶ ANSI date – DATE '2016-11-01'



# Метки

---

- ▶ Метка – способ именовать определенную часть программы
- ▶ Синтаксис << label>>
- ▶ Используется для:
  - ▶ Именования блока на время выполнения
  - ▶ Улучшение читаемости кода
  - ▶ Необходимость ссылаться во вложенном блоке на переменную с таким же именем из внешнего блока
  - ▶ Для перехода по GOTO



# Символы специального значения

---

;	Разделитель выражений
--	Однострочный и многострочный комментарий
/* abcd */	
% и _	Множественный и одиночный групповой символ в Like
@	Индикатор удаленного объекта
<>	Не равно
!=	
^=	
~=	
	Конкатенация
<<label>>	Метка
<= и =>	Меньше или равно / Больше или равно
:=	Присваивание



# Типы данных ORACLE - символьные

CHAR	Символьное поле фиксированной длины до 2000 байт
NCHAR	Поле фиксированной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 2000 символов или 2000 байт в зависимости от набора символов.
VARCHAR2	Символьное поле переменной длины до 4000 байт
NVARCHAR2	Поле переменной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 4000 символов или 4000 байт в зависимости от набора символов.

# Типы данных ORACLE – символьные

LONG	Символьный, переменной длины, до 2GB, оставлен для совместимости
RAW(n)	Переменной длины, для бинарных данных $n \leq 2000$ byte оставлен для совместимости
LONG RAW	Бинарные данные до 2GB
CLOB	Символьный тип большой объект до 4GB
NLOB	CLOB для многобайтных символов
BLOB	Большой двоичный объект до 4GB
BFILE	Указатель на двоичный файл операционной системы



# Типы данных ORACLE – дата/время

DATE	7 байтовое поле фиксированной длины, используемое для хранения даты и времени
INTERVAL DAY TO SECOND	11 байтовое поле фиксированной длины для интервала времени: Дни, часы, минуты, секунды
INTERVAL YEAR TO MONTH TIMESTAMP	5 байтовое поле фиксированной длины для интервала времени: Годы и месяцы
TIMESTAMP WITH TIME ZONE	13 байтовое поле фиксированной длины Дата, время и настройки, связанные с часовым поясом.
TIMESTAMP WITH LOCAL TIME	7-11 байтовое поле переменной длины Дата и время, приведенные к часовому поясу базы данных



# Типы данных ORACLE – числовые

NUMBER(n, s)

Числовой тип переменной длины

Точность n <= 38, общее количество цифр

Масштаб s = [-84,127], количество цифр после запятой

# Неявные преобразования типов данных

VARCHAR2 CHAR	DATE
DATE	VARCHAR2
VARCHAR2 CHAR	ROWID
ROWID	VARCHAR2
VARCHAR2 CHAR	NUMBER
NUMBER	VARCHAR2



# Character Set

---

- ▶ NLS - National Language Support, далее Globalization Support
- ▶ Можно хранить данные множества национальных языков, используя Unicode или специальные кодировки – наборы символов (**character set**)
- ▶ Символы хранятся как коды символов, зависящие от выбранного набора символов
- ▶ В одной БД могут использоваться два набора символов: основной (**database character set**) и дополнительный (**national character set**)
- ▶ Устанавливаются при создании БД
- ▶ Изменяются `alter database (national) character set`

# Поддержка национальных языков

---

- ▶ Основной набор символов используется для:
  - ▶ хранения символьных типов `char`, `varchar2`, `clob` и `long`
  - ▶ описания имен объектов, переменных
  - ▶ Ввода и хранения PL/SQL модулей
- ▶ Дополнительный набор символов используется для:
  - ▶ хранения символьных типов `nchar`, `nvarchar2`, `nclob`
- ▶ Кроме символов алфавита в набор включаются знаки препинания, числа, символы денежных единиц и пр.



# Поддержка национальных языков

- ▶ <region> <number of bits used to represent a character>  
<standard character set name> [S|C]

WE8ISO8859P1

Western European 8-bit ISO  
8859 Part 1

WE (Western Europe)

8



# Поддержка национальных языков

---

- ▶ Переменная окружения **NLS\_LANG**:
- ▶ **NLS\_LANG = language\_territory.charset**
  - ▶ Язык (**LANGUAGE**) – имена месяцев, имена дней, направление текста, сокращения для времени и дат. По умолчанию **AMERICAN**
  - ▶ Территория (**TERRITORY**) – настройки календаря, формат даты, формат денежной единицы. Если не указан, то будет взято значение, соответствующее языку (для **RUSSIAN - CIS**)
  - ▶ Набор символов (**CHARACTER SET**) – отображение символов, отображение и конвертация заглавных букв, порядок замещения символов при преобразовании. Каждому языку поставлен в соответствие набор символов по умолчанию



# Поддержка национальных языков

---

- ▶ Представления словаря:
- ▶ **NLS\_SESSION\_PARAMETERS**
- ▶ **NLS\_INSTANCE\_PARAMETERS**
- ▶ **NLS\_DATABASE\_PARAMETERS**



# Семантика символов

---

- ▶ Байтовая семантика рассматривает строки как последовательность байтов
- ▶ Символьная семантика рассматривает строки как последовательность символов
- ▶ Задается параметром **NLS\_LENGTH\_SEMANTICS**
- ▶ По умолчанию - BYTE
- ▶ Можно задавать семантику для столбца:
  - ▶ VARCHAR2(20 BYTE)
  - ▶ VARCHAR2(10 CHAR)



# Семантика символов

sh1 - SYSTEM - 09.11.10

SYSTEM@sh1> show parameter nls

NAME_COL_PLUS_SHOW_PARAM	TYPE	VALUE_CO
nls_calendar	string	
nls_comp	string	
nls_currency	string	
nls_date_format	string	
nls_date_language	string	
nls_dual_currency	string	
nls_iso_currency	string	
nls_language	string	AMERICAN
nls_length_semantics	string	BYTE
nls_nchar_conv_excp	string	FALSE
nls_numeric_characters	string	
NAME_COL_PLUS_SHOW_PARAM	TYPE	VALUE_CO
nls_sort	string	
nls_territory	string	AMERICA
nls_time_format	string	
nls_timestamp_format	string	
nls_timestamp_tz_format	string	
nls_time_tz_format	string	

SYSTEM@sh1>

# Типы данных PL/SQL

---

- ▶ Скалярные (Scalar)
- ▶ Ссылочные (Reference)
- ▶ Составные (Composite)
- ▶ Большие объекты (LOB)



# Скалярные типы данных

---

- ▶ символ/строка
- ▶ число
- ▶ булев
- ▶ дата/время



# Символы/строки

```
-- 06.sql
-- символы/строки
declare
    c1 char (5);          -- символьный тип фиксированной длины 5 байт
    c2 char (5 char);   -- символьный тип фиксированной длины 5 символов
    c3 nchar (5);        -- национальный символьный тип фиксированной длины 5 байт
    c4 nchar (5 char); -- национальный символьный тип фиксированной длины 5 символов

begin
    dbms_output.put_line('c1 ='||c1||', length(c1)='|| length(c1)); -- размер строки в символах
    dbms_output.put_line('c2 ='||c2||', length(c2)='|| length(c2));
    dbms_output.put_line('c3 ='||c3||', length(c3)='|| length(c3));
    dbms_output.put_line('c4 ='||c4||', length(c4)='|| length(c4));
    c1 := 'hhh';
    c2 := 'ммм';
    c3 := 'hhh';
    c4 := 'ммм';
    dbms_output.put_line('c1 ='||c1||', length(c1)='|| length(c1));
    dbms_output.put_line('c2 ='||c2||', length(c2)='|| length(c2));
    dbms_output.put_line('c3 ='||c3||', length(c3)='|| length(c3));
    dbms_output.put_line('c4 ='||c4||', length(c4)='|| length(c4));

end;
/
```

## Символы/строки

```

-- 07.sql
-- символы/строки
declare
    c1 varchar (15);          -- симв. SQL-типа (ANSI) переменной длины 15 байт max 32К (БД:4К)
    c2 varchar (15 char);     -- симв. SQL-типа (ANSI) переменной длины 15 символов
    c3 varchar2 (15);         -- синоним varchar
    c4 nvarchar2 (15 char);   -- национальный симв. тип переменной длины 15 символов

begin
    dbms_output.put_line('c1 ='||c1||', length(c1)='|| length(c1)); -- размер строки в символах
    dbms_output.put_line('c2 ='||c2||', length(c2)='|| length(c2));
    dbms_output.put_line('c3 ='||c3||', length(c3)='|| length(c3));
    dbms_output.put_line('c4 ='||c4||', length(c4)='|| length(c4));
    c1 := '1234567';
    c2 := '1234567';
    c3 := '1234567';
    c4 := '1234567';
    dbms_output.put_line('c1 ='||c1||', length(c1)='|| length(c1));
    dbms_output.put_line('c2 ='||c2||', length(c2)='|| length(c2));
    dbms_output.put_line('c3 ='||c3||', length(c3)='|| length(c3));
    dbms_output.put_line('c4 ='||c4||', length(c4)='|| length(c4));
end;
/

```

# Символы/строки

```
CREATE TABLE TEST (T1 CHAR(3), T2 VARCHAR2(3));
INSERT INTO TEST VALUES ('','');
SELECT T1, T2 FROM TEST WHERE T1 IS NULL;
SELECT T1, T2 FROM TEST WHERE T2 IS NULL;
```

T1	T2
1 (null)	(null)

```
DECLARE VCHR VARCHAR2(3) := '';
BEGIN
  IF VCHR IS NULL THEN DBMS_OUTPUT.PUT_LINE('VARCHAR2 TRUE');
  END IF;
END;

DECLARE CHR CHAR(3) := '';
BEGIN
  IF CHR IS NULL
    THEN DBMS_OUTPUT.PUT_LINE('CHAR TRUE');
    ELSE DBMS_OUTPUT.PUT_LINE('CHAR FALSE');
  END IF;
END;
```

```
VARCHAR2 TRUE
CHAR FALSE
```

# Числовые типы данных

```
-- 11/01.sql
-- number (n,s)      n=[0,38], s=[-84,127]
declare
    n1 number(20,11) := 123456789.12345678911;    --
    n2 number(20,10) := 123456789.12345678911;
    n3 number(20) := 123456789.12345678911;
    n4 number(20,-1) := 123456789.12345678911;
    n5 number(20,-2) := 123456789.12345678911;
    n6 number(20,-21) := 123456789.12345678911;
    n7 number(20,-25) := 123456789.12345678911;
    n8 number(20,-83) := 123456789.12345678911;
    n9 number(20,-7) := 0.12345678911;
    n10 number(38,10) := 123456789.12345678911;
    n11 number(38,10) := 123456789E-12;
begin
    dbms_output.put_line('n1 ='||n1);
    dbms_output.put_line('n2 ='||n2);
    dbms_output.put_line('n3 ='||n3);
    dbms_output.put_line('n4 ='||n4);
    dbms_output.put_line('n5 ='||n5);
    dbms_output.put_line('n6 ='||n6);
    dbms_output.put_line('n7 ='||n7);
    dbms_output.put_line('n8 ='||n8);
    dbms_output.put_line('n9 ='||n9);
    dbms_output.put_line('n10 ='||n10);
    dbms_output.put_line('n11 ='||n11);
end;
/

```

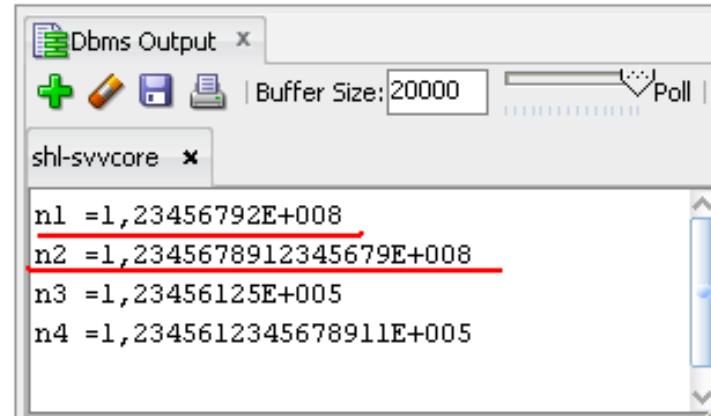
```
Dbms Output
+ - Buffer Size: 20000 Poll
shl-svvcore *
set serveroutput on
n1 =123456789,12345678911
n2 =123456789,1234567891
n3 =123456789
n4 =123456790
n5 =123456800
n6 =0
n7 =0
n8 =0
n9 =0
n10 =123456789,1234567891
n11 =,0001234568
```

# Числовые типы данных

```
-- 11/02.sql
| declare
|   n1 binary_float    := 123456789.12345678911;      -- IEEE-754
|   n2 binary_double   := 123456789.12345678911;      -- IEEE-754
|   n3 binary_float    := 123456.12345678911;
|   n4 binary_double   := 123456.12345678911;

begin
  dbms_output.put_line('n1 ='||n1);
  dbms_output.put_line('n2 ='||n2);
  dbms_output.put_line('n3 ='||n3);
  dbms_output.put_line('n4 ='||n4);
end;
/

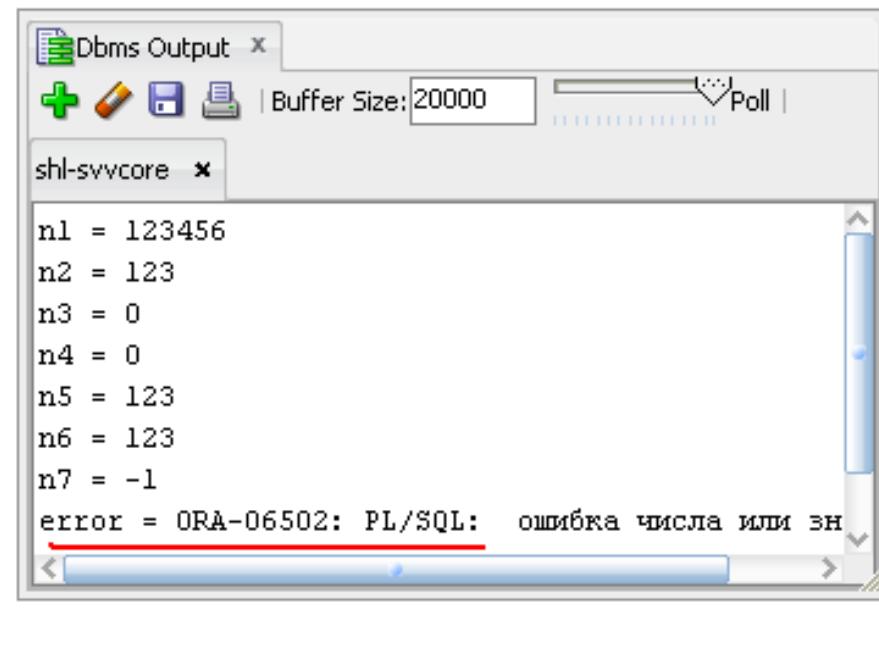
```



# Числовые типы данных

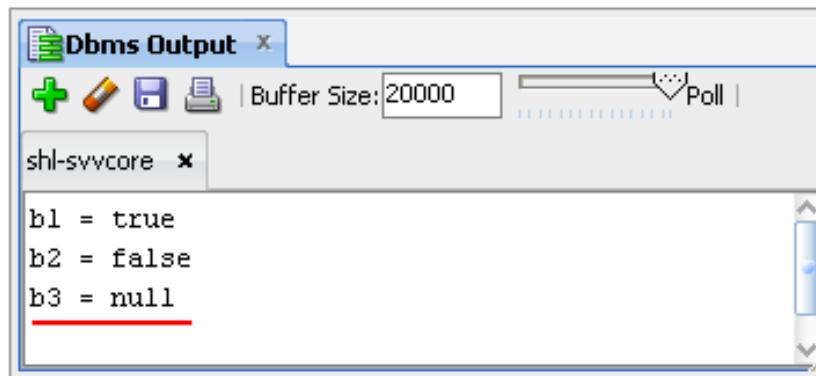
```
-- 11/03.sql
declare
    n1 binary_integer := 123456.12345678911;      -- [-2^31-1, 2^31-1] не БД
    n2 pls_integer := 123;                          -- [-2^31-1, 2^31-1] не БД
    n3 natural := 0;                                -- [0, 2^31-1] не БД
    n4 naturaln := 0;                               -- [0, 2^31-1] null не БД
    n5 positive := 123;                            -- [1,2^31-1] не БД
    n6 positiven := 123;                           -- [1,2^31-1] null не БД
    n7 signtype := - 0.5;                          -- ]-1,1[ ---> -1,0,+1
begin
    dbms_output.put_line('n1 = '||n1);
    dbms_output.put_line('n2 = '||n2);
    dbms_output.put_line('n3 = '||n3);
    dbms_output.put_line('n4 = '||n4);
    dbms_output.put_line('n5 = '||n5);
    dbms_output.put_line('n6 = '||n6);
    dbms_output.put_line('n7 = '||n7);
    n3 := -1;
exception
    when others
    then
        dbms_output.put_line('error = '||sqlerrm);
end;
/

```



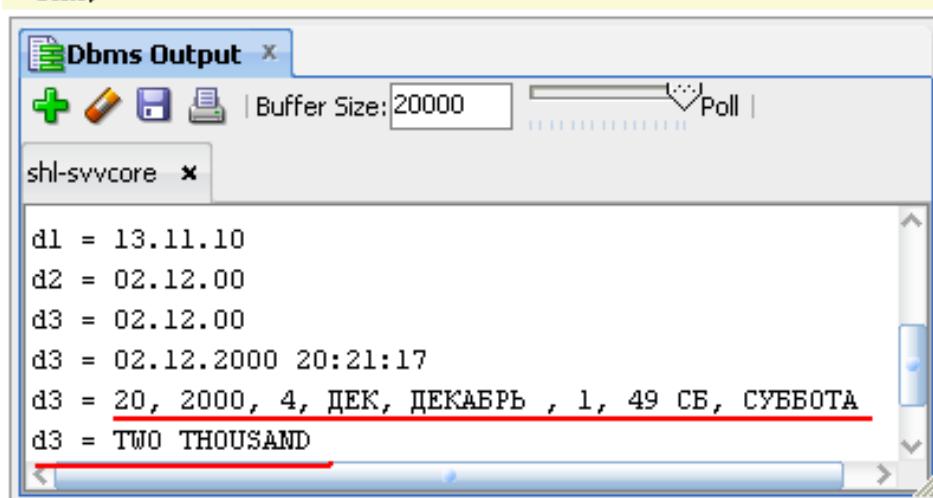
# Булев тип

```
-- 11/04.sql
declare
    b1 boolean := true;
    b2 boolean := false;
    b3 boolean ;
begin
    if b1      then dbms_output.put_line('b1 = ''true''); end if;
    if not b1     then dbms_output.put_line('b1 = ''false''); end if;
    if b2        then dbms_output.put_line('b2 = ''true''); end if;
    if not b2    then dbms_output.put_line('b2 = ''false''); end if;
    if b3        then dbms_output.put_line('b3 = ''true''); end if;
    if not b3    then dbms_output.put_line('b3 = ''false''); end if;
    if b3 is null then dbms_output.put_line('b3 = ''null''); end if;
end;
/
```



# Дата и время

```
-- 11/05.sql
declare
    d1 date := sysdate;
    d2 date := to_date('02.12.2000','DD.MM.YYYY') ;
    d3 date := to_date('02.12.2000 20:21:17','DD.MM.YYYY HH24:MI:SS') ;
begin
    dbms_output.put_line('d1 = '||d1);
    dbms_output.put_line('d2 = '||d2);
    dbms_output.put_line('d3 = '||d3);
    dbms_output.put_line('d3 = '||to_char(d3,'DD.MM.YYYY HH24:MI:SS'));
    dbms_output.put_line('d3 = '||to_char(d3,'CC, YYYY, Q, MON, MONTH, W, WW DY, DAY'));
    dbms_output.put_line('d3 = '||to_char(d3,'YEAR'));
end;
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. It contains a toolbar with icons for new connection, edit, save, and refresh, followed by a 'Buffer Size' input field set to 20000 and a 'Poll' button. Below the toolbar is a tab bar with 'shl-svvcore' selected. The main area displays the output of the PL/SQL block. The output consists of six lines of text:

```
d1 = 13.11.10
d2 = 02.12.00
d3 = 02.12.00
d3 = 02.12.2000 20:21:17
d3 = 20, 2000, 4, ДЕК, ДЕКАБРЬ , 1, 49 СБ, СУББОТА
d3 = TWO THOUSAND
```

# Дата и время

```
declare
  d1 timestamp(3) := systimestamp; -- [0,9] дробная часть секунд
  d2 timestamp(3) with time zone    := systimestamp;
  d3 timestamp(5) with local time zone  := systimestamp;
  d4 timestamp(5) with local time zone  := to_timestamp_tz ('02.12.2000 19:21:31 +02:00','DD.MM.YYYY HH24:MI:SS TZH:TZM');
  d5 timestamp(3) with time zone    := to_timestamp_tz('02.12.2000 19:21:31 EUROPE/MOSCOW MSK','DD.MM.YYYY HH24:MI:SS TZR TZD');

begin
  dbms_output.put_line('d1 = '||d1);
  dbms_output.put_line('d2 = '||d2);
  dbms_output.put_line('d3 = '||d3);
  dbms_output.put_line('d3 = '||to_char(d3,'DD.MM.YYYY HH24:MI:SS'));
  dbms_output.put_line('d3 = '||to_char(d3,'CC, YYYY, Q, MON, MONTH, W, WW DY, DAY, TZR, TZD'));
  dbms_output.put_line('d4 = '||to_char(d4,'DD.MM.YYY HH24:MI:SS'));
  dbms_output.put_line('d5 = '||to_char(d5,'DD.MM.YYYY HH24:MI:SS TZD'));
  dbms_output.put_line(to_char(sys_extract_utc(d5),'DD.MM.YYY HH24:MI:SS'));
  dbms_output.put_line(to_char(sys_extract_utc(d2),'DD.MM.YYY HH24:MI:SS'));
end;
/
```

The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output' and it has a buffer size of 20000. The output pane contains the following text:

```
d1 = 13-НОЯ-10 08.54.15,375 PM
d2 = 13-НОЯ-10 08.54.15,375 PM +02:00
d3 = 13-НОЯ-10 09.54.15,37500 PM
d3 = 13.11.2010 21:54:15
d3 = 21, 2010, 4, НОЯ, НОЯБРЬ , 2, 46 СБ, СУББОТА      , EUROPE/MOSCOW, MSK
d4 = 02.12.000 20:21:31
d5 = 02.12.2000 19:21:31 MSK
02.12.000 16:21:31
13.11.010 18:54:15
```

# Дата и время

---

```
SYSTEM@sh1> select * from sys.v$timezone_names  
2      where upper(tzname) = 'EUROPE/MINSK'  
3  /
```

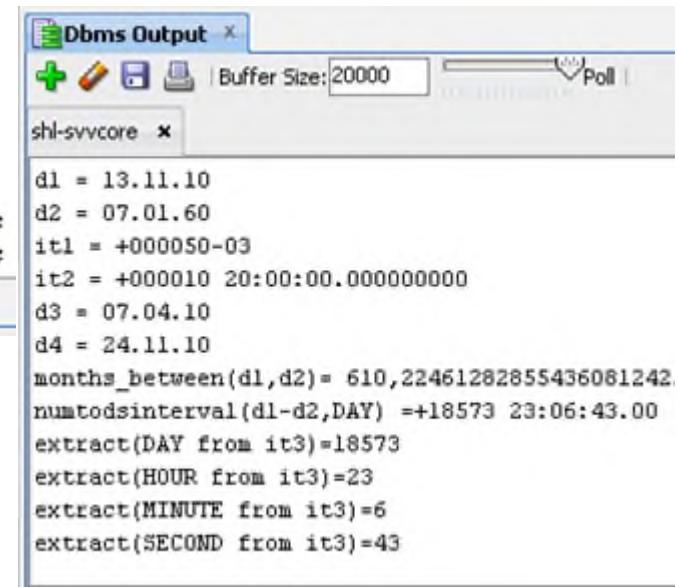
TZNAME	TZABBREV
Europe/Minsk	LMT
Europe/Minsk	MMT
Europe/Minsk	EET
Europe/Minsk	MSK
Europe/Minsk	CEST
Europe/Minsk	CET
Europe/Minsk	MSD
Europe/Minsk	EEST



# Дата и время

```
-- 11/09.sql
declare
    d1 date := sysdate;
    d2 date := to_date('07.01.1960','DD.MM.YYYY');
    it1 interval year(2) to month := interval '50-3' year to month;
    it2 interval day(5) to second(2) := interval '10 20:00:00' day to second ;
    d3 date := d2 + it1;
    d4 date := d1 + it2;
    it3 interval day(5) to second(2);
begin
    dbms_output.put_line('d1 ='||d1);
    dbms_output.put_line('d2 ='||d2);
    dbms_output.put_line('it1 ='||to_char(it1,'YY,MM'));
    dbms_output.put_line('it2 ='||to_char(it2,'YY,MM'));
    dbms_output.put_line('d3 ='||d3);
    dbms_output.put_line('d4 ='||d4);
    dbms_output.put_line('months_between(d1,d2)='||months_between(d1,d2));
    it3 := numtodsinterval(d1-d2,'DAY');
    dbms_output.put_line('numtodsinterval(d1-d2,DAY) ='||it3 );
    dbms_output.put_line('extract(DAY from it3)'||extract(DAY from it3));
    dbms_output.put_line('extract(HOUR from it3)'||extract(HOUR from it3));
    dbms_output.put_line('extract(MINUTE from it3)'||extract(MINUTE from it3));
    dbms_output.put_line('extract(SECOND from it3)'||extract(SECOND from it3));
end;
/

```



```
Dbms Output x
+ - Buffer Size: 20000 Poll
shl-svvcore x
d1 = 13.11.10
d2 = 07.01.60
it1 = +0000050-03
it2 = +0000010 20:00:00.0000000000
d3 = 07.04.10
d4 = 24.11.10
months_between(d1,d2)= 610,22461282855436081242
numtodsinterval(d1-d2,DAY) =+18573 23:06:43.00
extract(DAY from it3)=18573
extract(HOUR from it3)=23
extract(MINUTE from it3)=6
extract(SECOND from it3)=43
```

# Константы

```
1 -- 11/10.sql
2 declare
3     n1 constant number (5):=5;
4     d1 constant date := sysdate;
5     n2 pls_integer not null := 25;
6     v1 varchar(25) not null default 'Hello World';
7 begin
8     n1 := 10;
9 exception
10    when others
11        then dbms_output.put_line('error = '|| n1); | Error starting at line 2 in command:
12 end; | declare
13 / |     n1 constant number (5):=5;
14 |     d1 constant date := sysdate;
15 |     n2 pls_integer not null := 25;
16 |     v1 varchar(25) not null default 'Hello World';
17 | begin
18 |     n1 := 10;
19 | exception
20 |     when others
21 |         then dbms_output.put_line('error = '|| n1);
22 | end;

24 Error report:
25 ORA-06550: Стока 7, столбец 5:
26 PLS-00363: выражение 'N1' не м.б. использовано как адресат назначения
27 ORA-06550: Стока 7, столбец 5:
28 PL/SQL: Statement ignored
29 06550. 00000 - "line %s, column %s:\n%s"
30 *Cause: Usually a PL/SQL compilation error.
31 *Action:
```

# Связанные объявления переменных

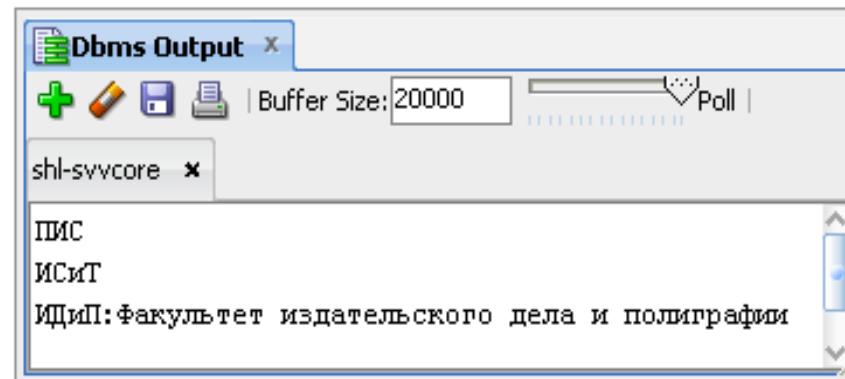
---

- ▶ Тип переменной основан на известной структуре данных
- ▶ Скалярная ссылка %TYPE для определения переменной на основе другой переменной или поля в таблице
- ▶ Ссылка на запись %ROWTYPE для определения структуры записи на основе таблицы или курсора



# %TYPE и %ROWTYPE

```
-- 11/11.sql
declare
    subject      svvcore.subject.subject%type;
    pulpit       svvcore.pulpit.pulpit%type;
    faculty_rec svvcore.faculty%rowtype;
begin
    subject := 'ПИС';
    pulpit := 'ИСиТ';
    faculty_rec.faculty := 'ИДиП';
    faculty_rec.faculty_name := 'Факультет издательского дела и полиграфии';
    dbms_output.put_line(subject);
    dbms_output.put_line(pulpit);
    dbms_output.put_line(rtrim(faculty_rec.faculty)||':'||faculty_rec.faculty_name);
exception
    when others
        then dbms_output.put_line('error = '|| sqlerrm);
end;
/
```



# Constraints

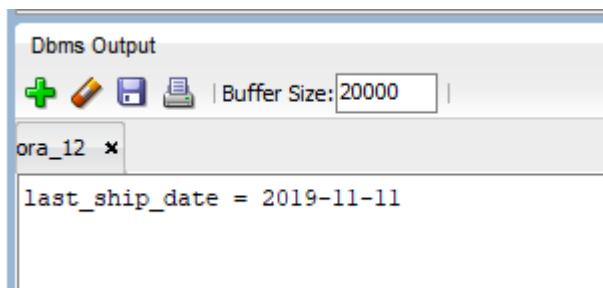
```
----- not null assignment
create table t11 (x date not null);

-- pl/sql var
DECLARE
    max_available_date DATE NOT NULL :=
        LAST_DAY(ADD_MONTHS(SYSDATE, 3));
    last_ship_date max_available_date%TYPE; -- not null variable
begin
    null;
end;

-- from table
DECLARE
    last_ship_date t11.x%TYPE; -- not null variable
begin
    last_ship_date:= to_date('2019-11-11','YYYY-MM-DD');
    dbms_output.put_line('last_ship_date = '||to_char(last_ship_date, 'YYYY-MM-DD'));
end;
```

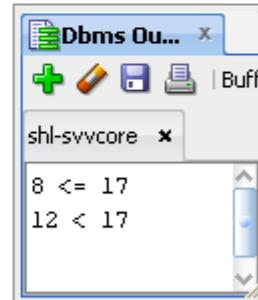
# Constraints

```
Error starting at line : 5 in command -
DECLARE
    max_available_date DATE NOT NULL :=
        LAST_DAY(ADD_MONTHS(SYSDATE, 3));
    last_ship_date max_available_date%TYPE; -- not null variable
begin
    null;
end;
Error report -
ORA-06550: Страна 4, столбец 20:
PLS-00218: a variable declared NOT NULL must have an initialization assignment
06550. 00000 - "line %s, column %s:\n%s"
*Cause: Usually a PL/SQL compilation error.
*Action:
```



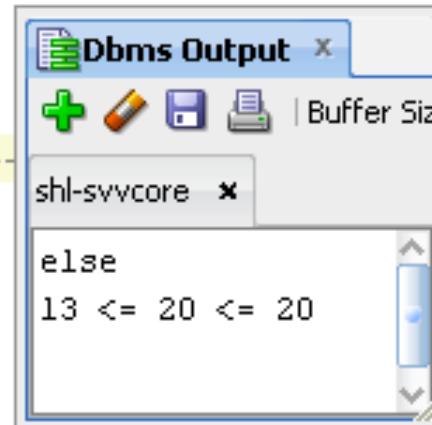
# Оператор IF

```
-- 117107541
declare
    x pls_integer := 17;
begin
    if 8 > x
    then
        dbms_output.put_line('8 > '||x);
    end if;
    -----
    if 8 > x
    then
        dbms_output.put_line('8 > '||x);
    else
        dbms_output.put_line('8 <= '||x);
    end if;
    -----
    if 8 > x
    then
        dbms_output.put_line('8 > '||x);
    elsif 8 = x
    then
        dbms_output.put_line('8 = '||x);
    elsif 12 > x
    then
        dbms_output.put_line('12 > '||x);
    elsif 12 = x
    then
        dbms_output.put_line('12 = '||x);
    else
        dbms_output.put_line('12 < '||x);
    end if;
end;
/
```



# Оператор CASE

```
-- 11/19.sql
declare
    x pls_integer := 17;
begin
-----
    case x
        when 1 then    dbms_output.put_line('1');
        when 2 then    dbms_output.put_line('2');
        when 3 then    dbms_output.put_line('3');
        else          dbms_output.put_line('else');
    end case;
-----
    case
        when 8 > x  then  dbms_output.put_line('8 > '||x);
        when 8 = x   then  dbms_output.put_line('8 = '||x);
        when 12 = x  then  dbms_output.put_line('12 = '||x);
        when x between 13 and 20 then dbms_output.put_line('13 <= '||x||' <= 20');
        else          dbms_output.put_line('else');
    end case;
-----
end;
```



# Циклы loop, for, while

```
-- 11/14.sql
declare
    x pls_integer := 0;
begin
    -----
    loop
        x := x + 1;
        dbms_output.put_line(x);
    exit when x > 5;
    end loop;
    -----
    for k in 1..5
    loop
        dbms_output.put_line(k);
    end loop;
    -----
    while (x > 0)
    loop
        x := x - 1;
        dbms_output.put_line(x);
    end loop;
    -----
end;
/
```

The screenshot shows the Oracle SQL Developer interface with two windows. The top window is titled 'Dbms Out...' and contains the text 'shl-svvcore'. Below it is a scrollable text area displaying the following output:

```
1
2
3
4
5
6
1
2
3
4
5
5
5
4
3
3
2
1
0
```

# TO\_NUMBER()

```
declare
a number;
begin
a := TO_NUMBER('F123.456,78',
                'L999G999D99',
                'NLS_NUMERIC_CHARACTERS='',.''
                || ' NLS_CURRENCY=''F''
                || ' NLS_ISO_CURRENCY=FRANCE');

dbms_output.put_line('a = '||a);
end;
```

```
a = 123456,78
```

```
SELECT TO_NUMBER('-AusDollars100','L9G999D99',
  'NLS_NUMERIC_CHARACTERS = ''.,'''
  'NLS_CURRENCY           = ''AusDollars''
  ') "Amount"
  FROM DUAL;
```

Amount	
1	-100

# TO\_CHAR()

```
----- С минус и плюс
select to_char(-1234567890, '99999999999S')
      from dual;
```

```
select to_char(1234567890, '99999999999S')
      from dual;
```

```
----- 99.99 десятичная точка
select to_char(0, '99.99')
      from dual;
```

```
select to_char(0.1, '99.99')
      from dual;
```

```
select to_char(-0.1, '99.99')
      from dual;
```

```
----- 090.990 - начальные и конечные нули
```

```
select to_char(0, '090.990')
      from dual;
```

```
select to_char(0.1, '090.990')
      from dual;
```

```
select to_char(-0.1, '090.990')
      from dual;
```

```
select to_char(-100, '090.990')
      from dual;
```

```
----- С минус и плюс
TO_CHAR(-1234567890,'99999999999S')
1 1234567890-
```

```
TO_CHAR(1234567890,'99999999999S')
1 1234567890+
```

```
----- Десятичные точки
TO_CHAR(0,'99.99')
1 .00
```

```
TO_CHAR(0.1,'99.99')
1 .10
```

```
----- Начальные и конечные нули
select to_char(-100, '090.990')
      from dual;
```

```
TO_CHAR(0,'090.990')
1 000.000
```

```
TO_CHAR(0.1,'090.990')
1 000.100
```

```
TO_CHAR(-0.1,'090.990')
1 -000.100
```

```
TO_CHAR(-100,'090.990')
1 -100.000
```

# TO\_CHAR()

----- 999,999.99 - разделитель

```
select to_char(-1000, '999,999.99')
  from dual;
```

----- В - пробелы по формату

```
select to_char(-0.1000, 'B999,999.99')
  from dual;
```

----- С - знак валюты  
----- NLS\_ISO\_CURRENCY

```
select to_char(10.10, 'C999,999.99')
  from dual;
```

TO_CHAR(10.10,'C999,999.99')	
1	RUB10.10

TO_CHAR(-0.1000,'B999,999.99')	
1	-.10

TO_CHAR(10.10,'C999,999.99')	
1	RUB10.10



# TO\_CHAR()

----- D - знак разделителя целой и дробной части

----- (NLS\_NUMERIC\_CHARACTER)

```
select to_char(10.10, 'C999D999')
      from dual;
```

TO_CHAR(10.10,'C999D999')	
1	RUB10,100

----- EEEE - знак научной записи

```
select to_char(10.10, '999EEEE')
      from dual;
```

TO_CHAR(10.10,'999EEEE')	
1	1E+01

----- G - знак разделителя

----- (NLS\_NUMERIC\_CHARACTER)

```
select to_char(100.100, '9G99')
      from dual;
```

TO_CHAR(100.100,'9G99')	
1	1 00

----- L - знак валюты

----- (NLS\_CURRENCY)

```
select to_char(100.100, 'L999.99')
      from dual;
```

TO_CHAR(100.100,'L999.99')	
1	?100.10

# TO\_CHAR()

```
----- MI - знак числа  
select to_char(-100.100, '999.99MI')  
  from dual;
```

TO_CHAR(-100.100,'999.99MI')	
1	100.10-

----- PR - разное отображение  
-- положительных (с пробелом)  
-- отрицательных / (в угловых скобках)  
select to\_char(100.100, '999.99PR')  
 from dual;

TO_CHAR(100.100,'999....')	
1	100.10

```
----- TM - минимально возможное число символов  
select to_char(100.100000, 'TM')  
  from dual;
```

TO_CHAR(100.100000...)	
1	100,1

----- U - знак валюты  
----- (NLS\_DUAL\_CURRENCY)  
select to\_char(100.100, 'U999.99')  
 from dual;

TO_CHAR(100.100,'U99...')	
1	p.100.10

----- V - \*10 - умножить  
select to\_char(100.100, '999V99')  
 from dual;

TO_CHAR(100.100,'99...')	
1	10010

----- XXXX - шестнадцатеричное значение (целое)  
-- только положительные  
select to\_char(100.100, 'XXXX')  
 from dual;

TO_CHAR(100.100,'XXXX')	
1	64

# Вопросы?

---



# **ORACLE 12c**

**PL/SQL**

**Лекция 11**

# Записи

---

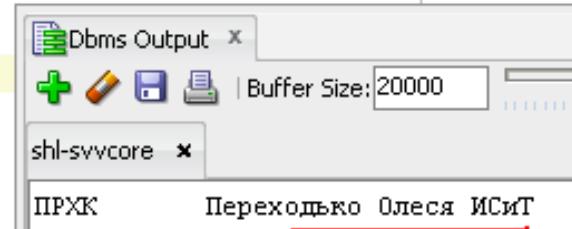
- ▶ Запись – структура данных, составленная из нескольких частей информации, называемых полями
- ▶ Для объявления записи вначале надо определить как тип, а потом объявить переменную типа «запись»
- ▶ Типы записей:
  - ▶ Табличные
  - ▶ Курсорные
  - ▶ Программно-определенные



# Записи

```
-- 13/04.sql
declare
    recl teacher%rowtype;
begin
    recl.teacher := 'ПРХК';
    recl.teacher_name := 'Переходько Олеся';
    recl.pulpit := 'ИСиТ';
    dbms_output.put_line(recl.teacher||' '||recl.teacher_name||' '||recl.pulpit);

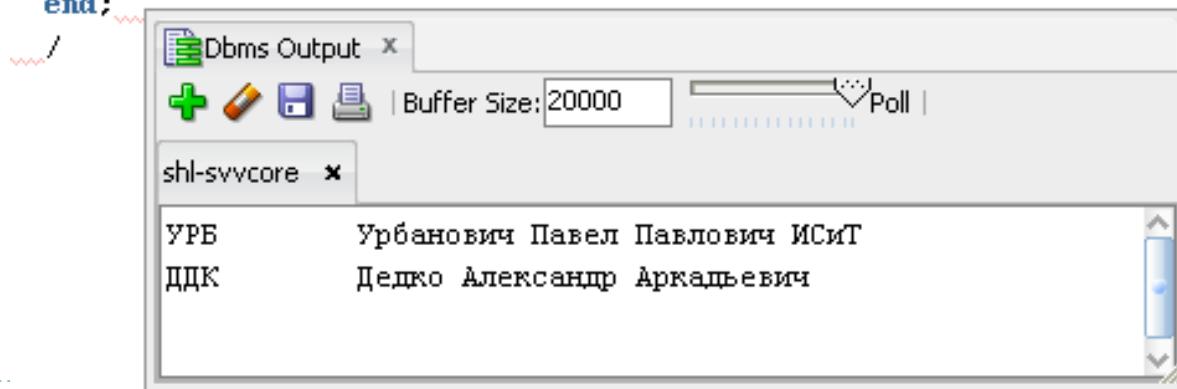
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output' and it has a buffer size of 20000. The output pane contains the following text:  
ПРХК      Переходько Олеся ИСиТ

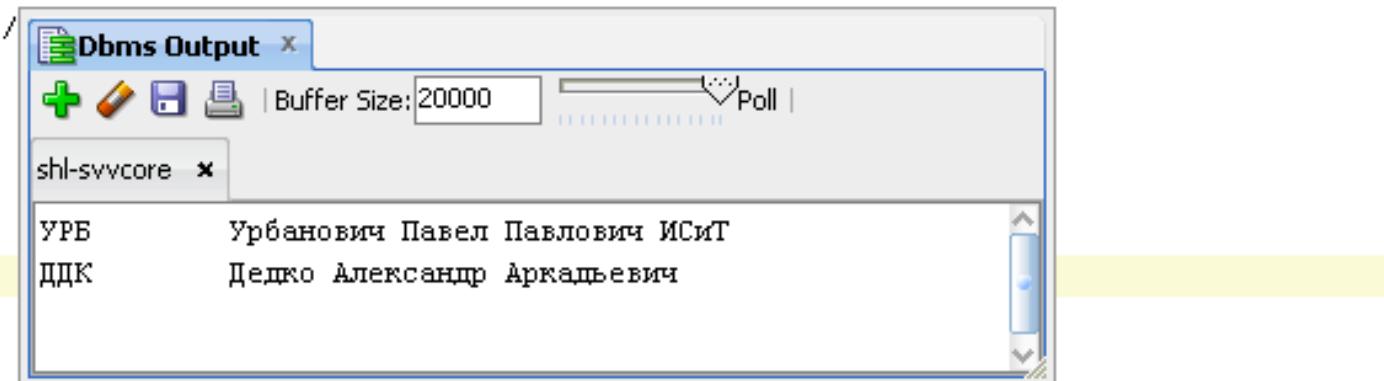
# Записи

```
-- 13/05.sql
declare
    recl teacher%rowtype;
    type person is record
    (
        code char(10),
        name varchar2(100)
    );
    rec2 person;
begin
    select * into recl from teacher where teacher = 'УРБ';
    select teacher, teacher_name into rec2 from teacher where teacher = 'ДДК';
    dbms_output.put_line(recl.teacher||' '||recl.teacher_name||' '||recl.pulpit);
    dbms_output.put_line(rec2.code||' '||rec2.name);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
```



# Записи

```
-- 15/05/2014
declare
    recl teacher%rowtype;
    type person is record
        (
            code teacher.teacher%type,
            name teacher.teacher_name%type
        );
    rec2 person;
begin
    select * into recl from teacher where teacher = 'УРБ';
    select teacher, teacher_name into rec2 from teacher where teacher = 'ДДК';
    dbms_output.put_line(recl.teacher||' '||recl.teacher_name||' '||recl.pulpit);
    dbms_output.put_line(rec2.code||' '||rec2.name);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. It contains two lines of output:

УРБ	Урбанович Павел Павлович ИСиТ
ДДК	Дедко Александр Аркаевич

# Использование полей записи

---

- ▶ Сравнение производится по полям записи
- ▶ Присвоение:
  - ▶ Присвоение для отдельного поля
  - ▶ `SELECT INTO` в запись в целом или в отдельные поля
  - ▶ Присвоение значения одной записи другой записи – для одного и того же объявления `TYPE`



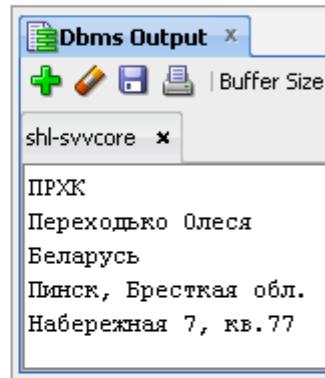
# Вложенные записи

```
-- 13/07.sql
declare
    recl teacher%rowtype;
    type address is record
    (
        address1 varchar2(100),
        address2 varchar2(100),
        address3 varchar2(100)
    );
    type person is record
    (
        code teacher.teacher%type,
        name teacher.teacher_name%type,
        homeaddress address
    );
    rec2 person;
begin
    rec2.code := 'ПРЖК';
    rec2.name := 'Переходько Олеся';
    rec2.homeaddress.address1 := 'Беларусь';
    rec2.homeaddress.address2 := 'Пинск, Брестская обл.';
    rec2.homeaddress.address3 := 'Набережная 7, кв.77';
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```

# Присваивание записей

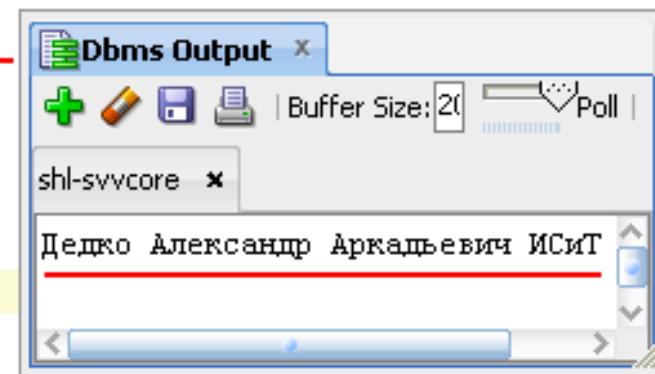
-- 13/10.sql

```
declare
  rec1 teacher%rowtype;
  type address is record
  (
    address1 varchar2(100),
    address2 varchar2(100),
    address3 varchar2(100)
  );
  type person is record
  (
    code teacher.teacher%type,
    name teacher.teacher_name%type,
    homeaddress address
  );
  rec2 person;
  rec3 person;
begin
  rec2.code := 'ПРХК';
  rec2.name := 'Переходъко Олеся';
  rec2.homeaddress.address1 := 'Беларусь';
  rec2.homeaddress.address2 := 'Минск, Брестская обл.';
  rec2.homeaddress.address3 := 'Набережная 7, кв.77';
  rec3 := rec2;
  dbms_output.put_line(rec3.code);
  dbms_output.put_line(rec3.name);
  dbms_output.put_line(rec3.homeaddress.address1);
  dbms_output.put_line(rec3.homeaddress.address2);
  dbms_output.put_line(rec3.homeaddress.address3);
exception
  when others then dbms_output.put_line(sqlerrm);
end;
```



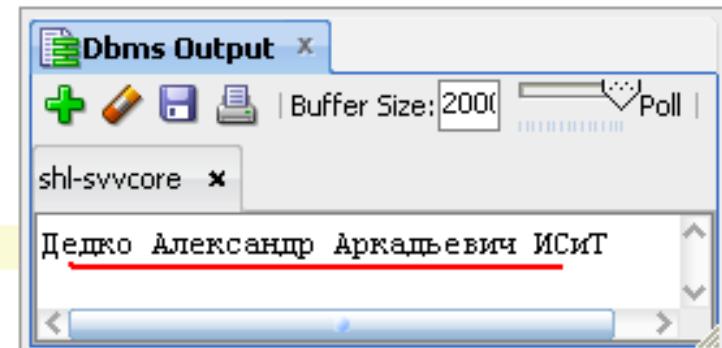
# Запись - параметр процедуры и функции

```
-- 13/11.sql
declare
    type tperson is record
    (
        name teacher.teacher_name%type,
        pulpit teacher.pulpit%type
    );
    rec tperson;
procedure gettperson (code teacher.teacher%type, r out tperson)
is
begin
    select teacher.teacher_name, teacher.pulpit into r
    from teacher
    where teacher = code;
end gettperson;
begin
    gettperson('ДДК', rec);
    dbms_output.put_line(rec.name || ' ' || rec.pulpit);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



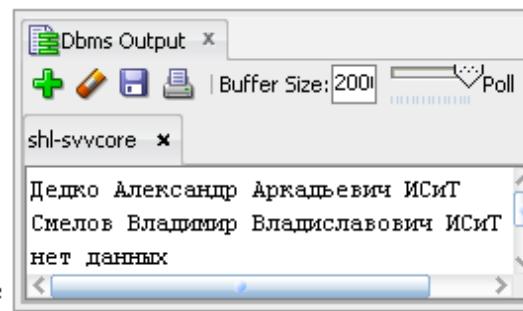
# Записи

```
-- 13/12.sql
declare
    type tperson is record
    (
        name    teacher.teacher_name%type,
        pulpit teacher.pulpit%type
    );
    rec tperson;
    function gettperson (code teacher.teacher%type) return tperson
    is
        r tperson;
    begin
        select teacher.teacher_name, teacher.pulpit into r
        from teacher
        where teacher = code;
        return r;
    end gettperson;
begin
    rec := gettperson('ДДК');
    dbms_output.put_line(rec.name||' '||rec.pulpit);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



# Записи

```
declare
  type tperson is record |
  (
    name teacher.teacher_name%type,
    pulpit teacher.pulpit%type
  );
  rec tperson;
  function gettperson (code teacher.teacher%type, r out tperson) return boolean
  is
    begin
      select teacher.teacher_name, teacher.pulpit into r
      from teacher
      where teacher = code;
      return true;
    exception
      when others then return false;
    end gettperson;
  begin
    if gettperson('ДДК', rec)
    then dbms_output.put_line(rec.name||' '||rec.pulpit);
    else dbms_output.put_line('нет данных');
    end if;
    if gettperson('СМНВ', rec)
    then dbms_output.put_line(rec.name||' '||rec.pulpit);
    else dbms_output.put_line('нет данных');
    end if;
    if gettperson('XXX', rec)
    then dbms_output.put_line(rec.name||' '||rec.pulpit);
    else dbms_output.put_line('нет данных');
    end if;
  exception
    when others then dbms_output.put_line(sqlerrm);
  end;
```



# Коллекции

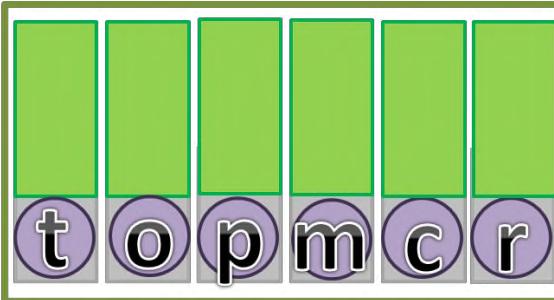
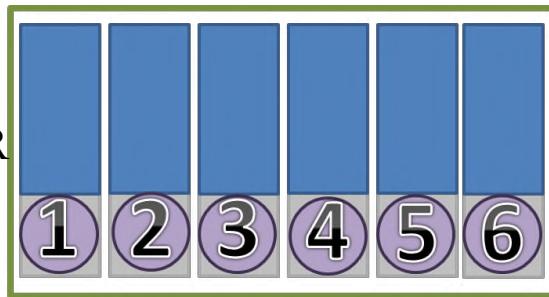
---

- ▶ Коллекция – структура данных, содержащая элементы одного типа
  - ▶ Элементом коллекции может быть как скалярная величина, так и композитные данные
  - ▶ Элементы коллекций можно сравнивать между собой на эквивалентность
  - ▶ Можно передавать параметром
  - ▶ Одномерная, но можно создавать коллекции коллекций
    - ▶ Массив переменной длины VARRAY
    - ▶ Вложенная таблица (nested tables)
    - ▶ Ассоциативный массив (associative array)
- 



# Collection Types

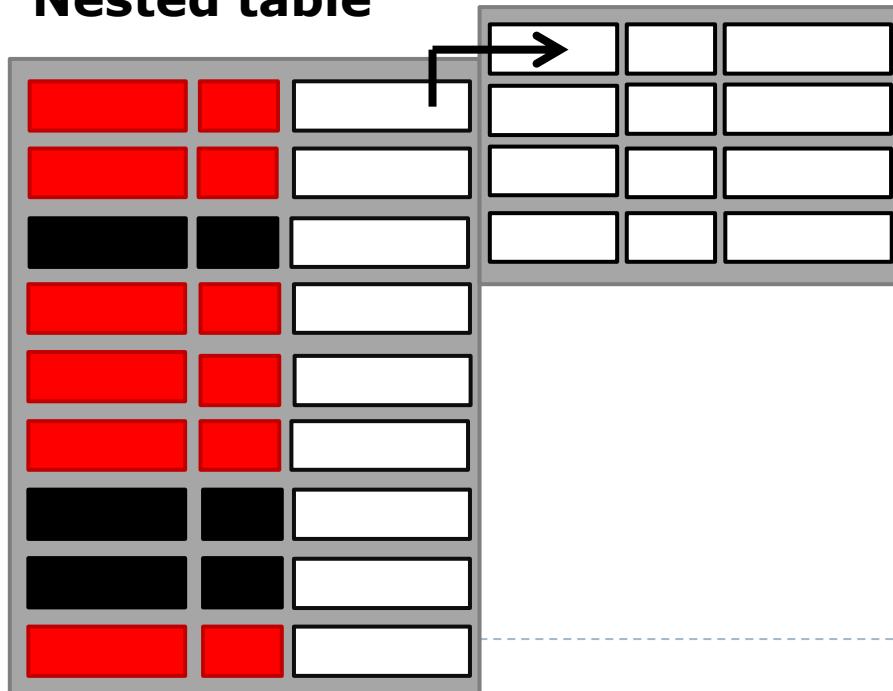
Index by  
PLS\_INTEGER



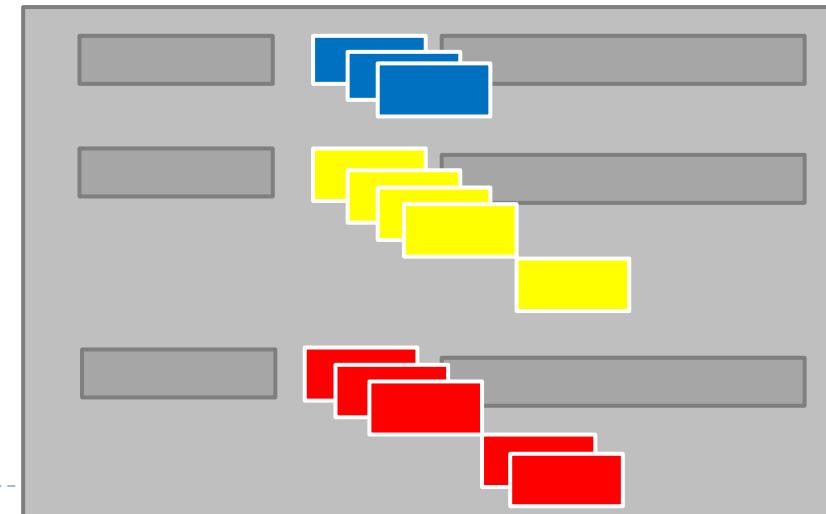
Index by  
VARCHAR

**Associative array**

**Nested table**



**Varray**



# Термины

---

- ▶ Коллекция состоит из **набора** элементов, причем каждый элемент находится в определенной позиции (имеется **индекс** элемента)
- ▶ Необходимо объявить **тип коллекции** – командой TYPE
- ▶ Необходимо объявить **коллекцию** – переменную этого типа для дальнейшего использования



# Ограниченнaя и неограниченaя коллекция

---

- ▶ Коллекция называется **ограниченной**, если заранее определены границы возможных значений индексов ее элементов, иначе **неограниченной**
- ▶ Коллекции типа VARRAY всегда ограничены
- ▶ Вложенные таблицы и ассоциативные массивы неограничены



# Разреженные и плотные коллекции

---

- ▶ Коллекция называется плотной, если все ее элементы, определены и каждому из них присвоено некоторое значение (таковым может быть и NULL)
- ▶ Массивы VARRAY всегда являются плотными
- ▶ Вложенные таблицы первоначально всегда плотные, но по мере удаления некоторых элементов становятся разреженными
- ▶ Ассоциативные массивы могут быть как разреженными, так и плотными в зависимости от способа их заполнения



# Работа с коллекциями

---

- ▶ Объявление коллекций
- ▶ Инициализация коллекций
  - ▶ Явно с помощью конструктора
  - ▶ Неявно при выборке из базы данных
  - ▶ Прямым присвоением переменной с другой коллекции такого же типа
- ▶ Добавление и удаление элементов
  - ▶ Вложенные таблицы и массивы переменной длины – сначала увеличить размер при помощи функции EXTEND, а затем присвоить значения новым элементам
  - ▶ Ассоциативный массив – присвоение значения новому элементу



# Массивы переменной длины

---

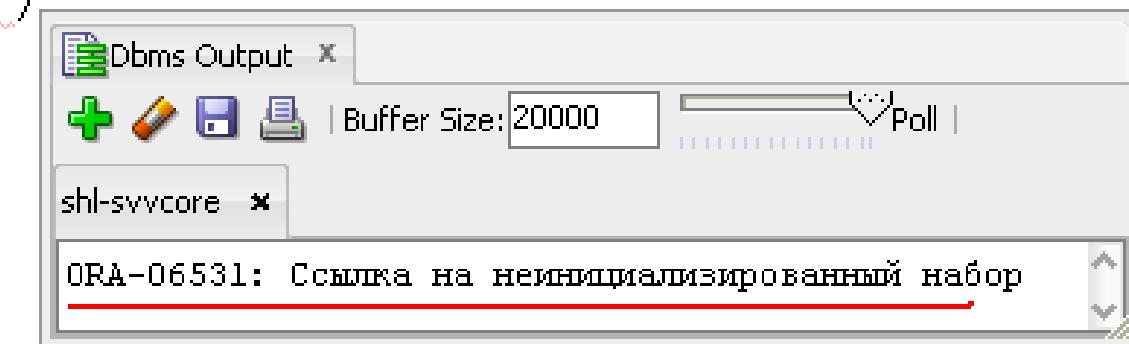
- ▶ Массивы переменной длины – одномерные, связанные коллекции однотипных элементов
- ▶ Доступны в рамках PL/SQL и в БД
- ▶ Являются плотными



# Массивы переменной длины

```
-- 13/14.sql
declare
    type myarraytype is varray(3) of number;
    va myarraytype;
begin
    for i in 1..3 loop
        va(i) := 1;
    end loop;

exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



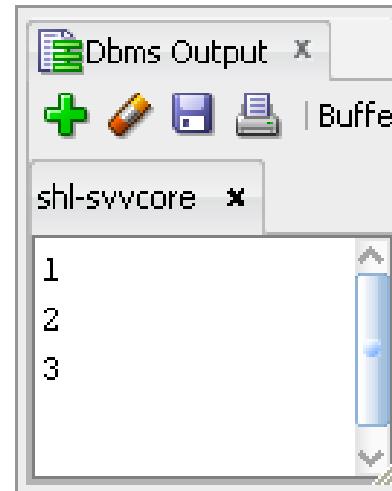
# Массивы переменной длины

```
-- 13/14.sql
```

```
declare
    type myarraytype is varray(3) of number;
    va myarraytype := myarraytype(null,null,null);

begin
    for i in 1..3
    loop
        va(i) := i;
    end loop;
    for i in 1..3
    loop
        dbms_output.put_line(va(i));
    end loop;

exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```

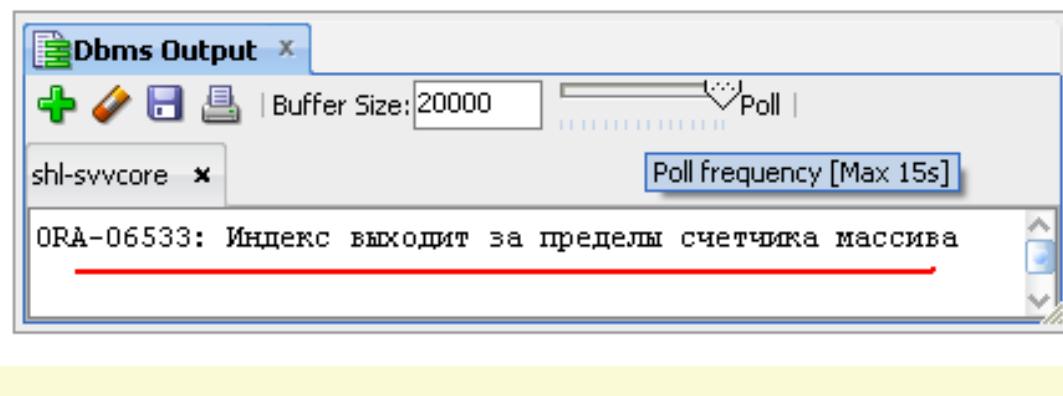


# Массивы переменной длины

```
-- 13/16.SQL
declare
    type myarraytype is varray(3) of number;
    va myarraytype := myarraytype(null, null);

begin
    for i in 1..3
    loop
        va(i) := i;
    end loop;
    for i in 1..3
    loop
        dbms_output.put_line(va(i));
    end loop;

exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



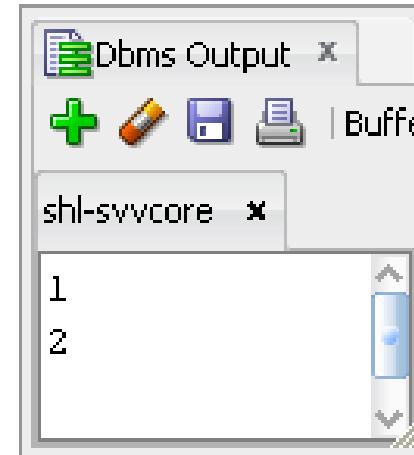
# Массивы переменной длины

-- 13/17.sql

```
declare
    type myarraytype is varray(3) of number;
    va myarraytype := myarraytype(null,null);

begin
    for i in 1..va.count
    loop
        va(i) := i;
    end loop;
    for i in 1..va.count()
    loop
        dbms_output.put_line(va(i));
    end loop;

```



```
exception
    when others then dbms_output.put_line(sqlerrm);
end;
```



# Массивы переменной длины

```
-- 13/18.sql
declare
    type myarraytype  is varray(3) of number not null;
    va myarraytype := myarraytype();

begin
    dbms_output.put_line('va.count = '||va.count);
    va.extend;

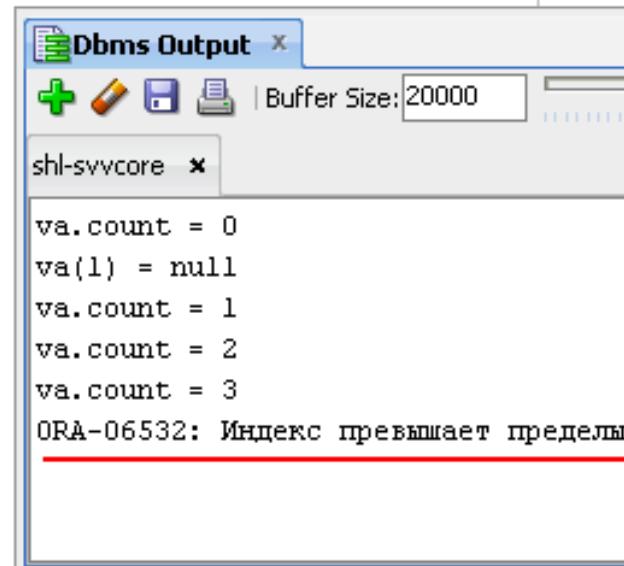
    if va(1) is null
    then dbms_output.put_line('va(1) = null '||va(1));
    else dbms_output.put_line('va(1) = '||va(1));
    end if;
    dbms_output.put_line('va.count = '||va.count);

    va.extend;
    dbms_output.put_line('va.count = '||va.count);

    va.extend;
    dbms_output.put_line('va.count = '||va.count);

    va.extend;
    dbms_output.put_line('va.count = '||va.count);

exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



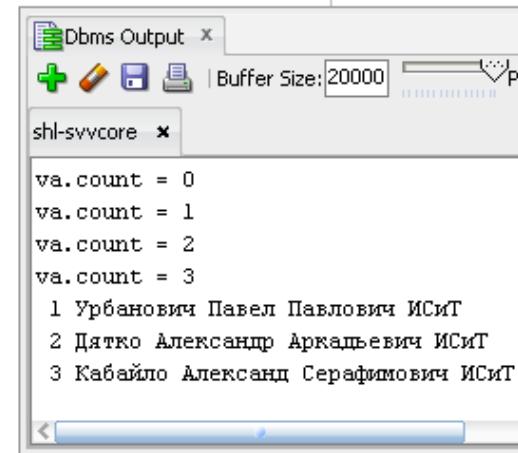
The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output' and it has a buffer size of 20000. The output pane contains the following text:

```
shl-svvcore x
va.count = 0
va(1) = null
va.count = 1
va.count = 2
va.count = 3
ORA-06532: Индекс превышает пределы
```

# Массивы переменной длины

```
-- 13/19.sql
declare
  type tperson is record
  (
    name teacher.teacher_name%type,
    pulpit teacher.pulpit%type
  );
  type myarraytype is varray(3) of tperson;
  va myarraytype := myarraytype();
  rec tperson; -- := tperson('xxx','xxxxxx');
begin
  dbms_output.put_line('va.count ='||va.count);

  va.extend;
  dbms_output.put_line('va.count ='||va.count);
  va(1).name := 'Урбанович Павел Павлович';
  va(1).pulpit := 'ИСиТ';
  va.extend;
  dbms_output.put_line('va.count ='||va.count);
  va(2).name := 'Дятко Александр Аркальевич';
  va(2).pulpit := 'ИСиТ';
  va.extend;
  dbms_output.put_line('va.count ='||va.count);
  va(3).name := 'Кабайло Александр Сергеевич';
  va(3).pulpit := 'ИСиТ';
  for i in 1..va.count
  loop
    dbms_output.put_line(' '||i||' '||va(i).name||' '||va(i).pulpit);
  end loop;
exception
  when others then dbms_output.put_line(sqlerrm);
end;
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. It contains two tabs: 'shl-svvcore' and 'x'. The 'shl-svvcore' tab displays the output of the PL/SQL script. The output shows the count of elements in the array and the details of each element (name and pulpit). The output is as follows:

```
va.count = 0
va.count = 1
va.count = 2
va.count = 3
1 Урбанович Павел Павлович ИСиТ
2 Дятко Александр Аркальевич ИСиТ
3 Кабайло Александр Сергеевич ИСиТ
```

# Вложенные таблицы

---

- ▶ Вложенные таблицы – одномерные, несвязанные коллекции однотипных элементов
- ▶ Доступны в рамках PL/SQL и как поля таблицы в БД
- ▶ Изначально являются плотными, но могут впоследствии становиться разреженными



# Вложенные таблицы

```
-- 14/01.sql
declare
  type tperson is record
  (
    name teacher.teacher_name%type,
    pulpit teacher.pulpit%type
  );
  type mytable is table of tperson;
  tb mytable := mytable();
  rec tperson; -- := tperson('xxx','xxxxxx');
begin
  dbms_output.put_line('tb.count = '||tb.count);

  tb.extend;
  dbms_output.put_line('tb.count = '||tb.count);
  tb(1).name := 'Урбанович Павел Павлович';
  tb(1).pulpit := 'ИСиТ';
  tb.extend;
  dbms_output.put_line('tb.count = '||tb.count);
  tb(2).name := 'Дятко Александр Аркальевич';
  tb(2).pulpit := 'ИСиТ';
  tb.extend;
  dbms_output.put_line('tb.count = '||tb.count);
  tb(3).name := 'Кабайло Александър Серафимович';
  tb(3).pulpit := 'ИСиТ';
  for i in 1..tb.count
  loop
    dbms_output.put_line( ' '||i||' '||tb(i).name||' '||tb(i).pulpit);
  end loop;
exception
  when others then dbms_output.put_line(sqlerrm);
end;
/
```

```
Dbms Output
shl-svvcore
set serveroutput on
tb.count = 0
tb.count = 1
tb.count = 2
tb.count = 3
1 Урбанович Павел Павлович
2 Дятко Александр Аркальевич
3 Кабайло Александър Серафимович
```

# Ассоциативные массивы

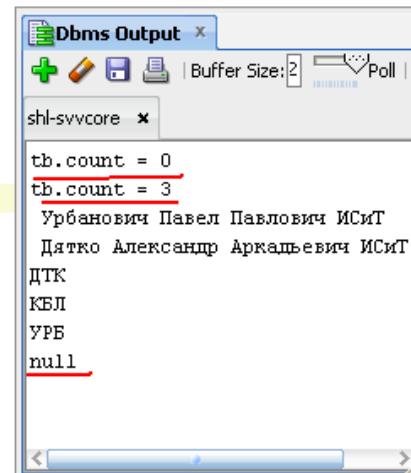
---

- ▶ Ассоциативные массивы – одномерные, неограниченные (по максимальному количеству элементов при создании) коллекции элементов
- ▶ Доступны только в рамках PL/SQL
- ▶ Изначально являются разреженными, индекс может принимать непоследовательные значения



# Ассоциативные массивы

```
-- 14/02.sql
declare
    type tperson is record (name teacher.teacher_name%type, pulpit teacher.pulpit%type );
    type mytable is table of tperson index by teacher.teacher_name%type;
    tb mytable;
    ntx teacher.teacher_name%type;
    rec tperson;
begin
    dbms_output.put_line('tb.count ='||tb.count);
    tb('УРБ').name := 'Урбанович Павел Павлович';
    tb('УРБ').pulpit := 'ИСиТ';
    tb('ДТК').name := 'Дятко Александр Аркашевич';
    tb('ДТК').pulpit := 'ИСиТ';
    tb('КЕЛ').name := 'Кабайло Александр Серафимович';
    tb('КЕЛ').pulpit := 'ИСиТ';
    dbms_output.put_line('tb.count ='||tb.count);
    dbms_output.put_line( '|||tb(''УРБ'').name|||'||tb(''УРБ'').pulpit);
    rec := tb('ДТК');
    dbms_output.put_line( '|||rec.name|||'||rec.pulpit);
    ntx := tb.first;
    if ntx is not null then dbms_output.put_line(ntx);
    else dbms_output.put_line('null');
    end if;
    ntx := tb.next(ntx);
    if ntx is not null then dbms_output.put_line(ntx);
    else dbms_output.put_line('null');
    end if;
    ntx := tb.next(ntx);
    if ntx is not null then dbms_output.put_line(ntx);
    else dbms_output.put_line('null');
    end if;
    ntx := tb.next(ntx);
    if ntx is not null then dbms_output.put_line(ntx);
    else dbms_output.put_line('null');
    end if;
exception
    when others then dbms_output.put_line(sqlerrm);
end;
```



# Методы и исключения коллекций

```
-- 14/03.sql
declare

begin
null;
-----methods -----
-- COUNT                      ALL
-- DELETE (n), DELETE (n,m)   ALL
-- EXISTS (n)                 ALL
-- EXTEND, EXTEND (n)         VARRAY, TABLE
-- FIRST                       ALL
-- LAST                        ALL
-- LIMIT                       VARRAY
-- NEXT(n)                     ALL
-- PRIOR(n)                    ALL
-- TRIM                         ALL
-- TRIM(n)                     ALL
----- exceptions -----
-- COLLECTION_IS_NULL          ALL
-- NO_DATA_FOUND                ALL
-- SUBSCRIPT_BEYOND_COUNT      VARRAY, TABLE
-- SUBSCRIPT_OUTSIDE_LIMIT     VARRAY, TABLE
-- VALUE_ERROR                  ALL
end;
/

```



# Сравнение характеристик коллекций

---

- ▶ Размерность?
  - ▶ Можно ли использовать как поле в таблице?
  - ▶ Неинициализированное состояние?
  - ▶ Инициализация?
  - ▶ Диапазон индексов?
  - ▶ Разреженность?
  - ▶ Ограничен по максимальному количеству элементов?
  - ▶ Можно ли присваивать значение любому элементу?
  - ▶ Метод расширения и уменьшения?
  - ▶ Можно ли сравнивать на равенство весь объект целиком?
  - ▶ Элементы сохраняют позицию при записи или чтении из БД?
-

# Вопросы?

---



# **ORACLE 12c**

**PL/SQL Курсоры**

**Лекция 10**

# Курсы

---

- ▶ **Курсор** — объект БД, который позволяет приложениям работать с записями построчно
- ▶ Поддерживаются два типа курсоров:
  - ▶ явный — объявляется разработчиком;
  - ▶ неявный — не требует объявления
- ▶ Курсор может возвращать одну строку, несколько строк или ни одной строки
- ▶ Для повторного создания результирующего набора для других значений параметров курсор следует закрыть, а затем повторно открыть



# Операторы управления явным курсором

---

- ▶ DECLARE — выполняет объявление явного курсора.
- ▶ OPEN — открывает курсор, создавая новый результирующий набор на базе указанного запроса.
- ▶ FETCH — выполняет последовательное извлечение строк из результирующего набора от начала до конца.
- ▶ CLOSE — закрывает курсор и освобождает занимаемые им ресурсы.



## Атрибуты курсора

---

- ▶ %ISOPEN — возвращает значение TRUE, если курсор открыт
- ▶ %FOUND — определяет, найдена ли строка, удовлетворяющая условию
- ▶ %NOTFOUND — возвращает TRUE, если строка не найдена
- ▶ %ROWCOUNT — возвращает номер текущей строки



# Курсыры

---

- ▶ Курсор Oracle – указатель на область в PGA, в которой хранится:
  - ▶ 1) строки запроса,
  - ▶ 2) число строк,
  - ▶ 3) указатель на разобранный запрос в общем пуле.
- ▶ Открытие курсора – создание контекстной области PGA – создается моментальный снимок (snapshot) данных запроса.



# Курсыры

---

- ▶ PL/SQL позволяет создавать 2 вида курсоров:
  - ▶ 1) Статические курсоры, SQL выражение для которых определяется на этапе компиляции:
    - ▶ Используются для DML команд
    - ▶ Могут быть явно объявлены и именованы
  - ▶ 2) Динамические курсоры, SQL выражение для которых определяется на этапе выполнения:
    - ▶ Могут использоваться для любых SQL выражений, включая DDL и DCL команды
    - ▶ Реализуются с помощью оператора EXECUTE IMMEDIATE или пакета dbms\_sql



# Неявные курсоры

---

- ▶ Неявные курсор – выполнение SQL выражения в секции исполнения или в секции исключений блока
- ▶ Операторы **INSERT**, **UPDATE**, **DELETE**, **MERGE**, **SELECT INTO**
- ▶ Не требуют объявления
- ▶ Не требуют **OPEN**, **FETCH**, **CLOSE**



# Неявные курсоры

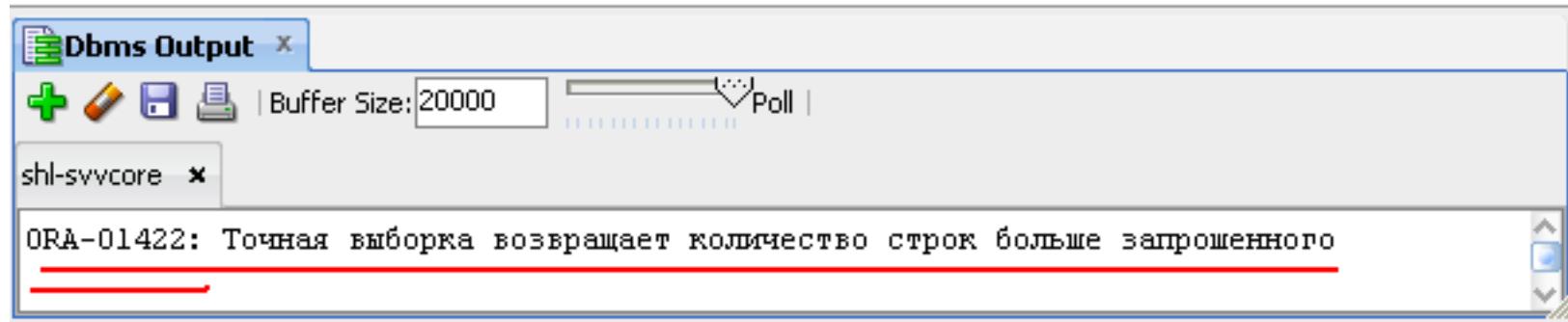
---

- ▶ Когда неявный курсор не возвращает строк вообще, PL/SQL генерирует исключение `NO_DATA_FOUND` и передает управление в секцию исключений
- ▶ Когда `SELECT` возвращает более одной строки, PL/SQL генерирует исключение `TOO_MANY_ROWS` и также передает управление в секцию исключений
- ▶ `SELECT INTO` предназначен исключительно для того, чтобы возвращать ровно 1 строку – точную выборку



# Неявные курсоры

```
-- 11/10.041
declare
    faculty_rec faculty%rowtype;
begin
    select * into faculty_rec from faculty;
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```

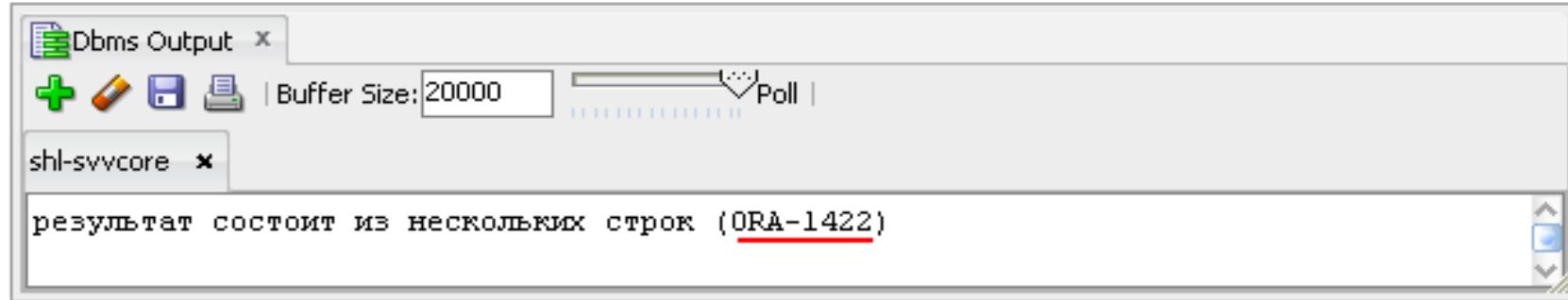


The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. It contains a toolbar with icons for new connection, edit, save, and refresh, followed by a 'Buffer Size: 20000' input field and a 'Poll' button. Below the toolbar, there is a tab bar with 'sh1-svvcore x'. The main area of the window displays the error message 'ORA-01422: Точная выборка возвращает количество строк больше запрошеннного' (ORA-01422: exact query returns more rows than requested). The entire message is highlighted with a red underline.

```
ORA-01422: Точная выборка возвращает количество строк больше запрошеннного
```

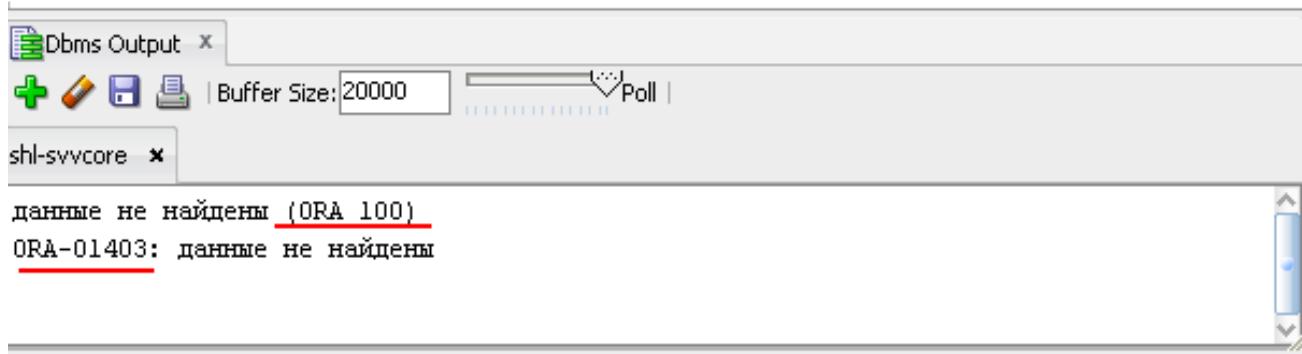
# Неявные курсоры

```
declare
    faculty_rec faculty%rowtype;
begin
    select * into faculty_rec from faculty;
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
exception
    when too_many_rows
        then dbms_output.put_line('результат состоит из нескольких строк (ORA'|| sqlcode||')');
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



# Неявные курсоры

```
-- 11/17.sql
declare
    faculty_rec faculty%rowtype;
begin
    select * into faculty_rec from faculty where faculty = 'XXX';
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
exception
    when no_data_found
    then
        dbms_output.put_line('данные не найдены (ORA ' || sqlcode || ')');
        dbms_output.put_line(sqlerrm);
    when too_many_rows
    then dbms_output.put_line('результат состоит из нескольких строк (ORA' || sqlcode || ')');
    when others
    then dbms_output.put_line(sqlerrm);
end;
/
```



# PRAGMA

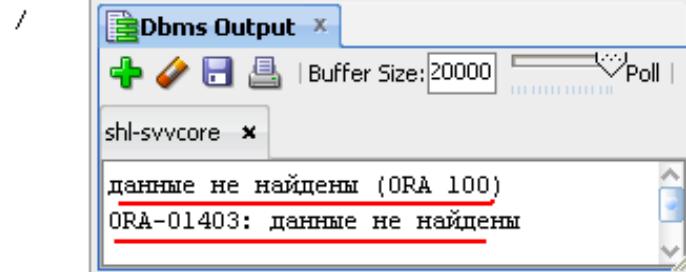
---

- ▶ Ключевое слово PRAGMA используется для того, чтобы указать директиву компилятору PL/SQL
- ▶ PRAGMA instruction
- ▶ Не транслируется в исполняемый код
- ▶ Разрешается использовать в секции декларации текущего блока



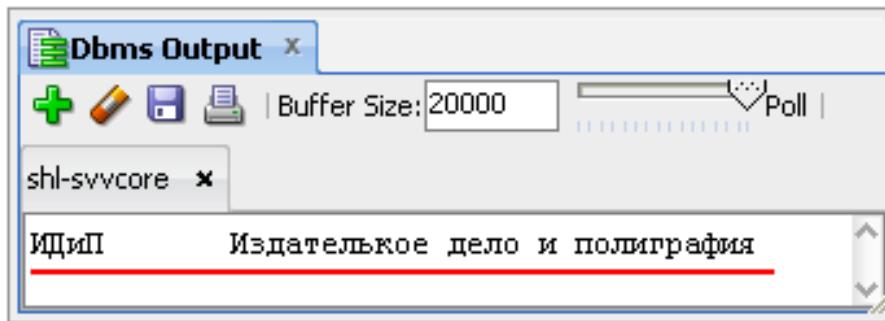
# Неявные курсоры – исключения

```
-- 11/17.sql
declare
    faculty_rec faculty%rowtype;
    my_exception exception;
    pragma exception_init(my_exception, 100);
begin
    select * into faculty_rec from faculty where faculty = 'XXX';
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
exception
    when my_exception
    then
        dbms_output.put_line('данные не найдены (ORA '||sqlcode||')');
        dbms_output.put_line(sqlerrm);
    --when no_data_found
    --then
    --    dbms_output.put_line('данные не найдены (ORA '||sqlcode||')');
    --    dbms_output.put_line(sqlerrm);
    when too_many_rows
    then dbms_output.put_line('результат состоит из нескольких строк (ORA'|| sqlcode||')');
    when others
    then dbms_output.put_line(sqlerrm);
end;
```



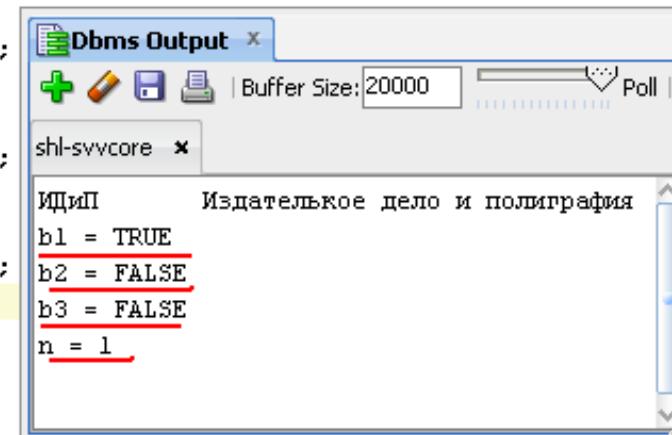
# Неявные курсоры

```
-- 11/18.sql
declare
    faculty_rec faculty%rowtype;
begin
    select * into faculty_rec from faculty where faculty = 'ИДиП';
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
exception
when no_data_found
then dbms_output.put_line('данные не найдены (ORA '||sqlcode||')');
when too_many_rows
then dbms_output.put_line('результат состоит из нескольких строк (ORA'|| sqlcode||')');
when others
then dbms_output.put_line(sqlerrm);
end;
/
```



# Неявные курсоры – атрибуты курсора

```
-- 11/30.sql
declare
    b1    boolean;
    b2    boolean;
    b3    boolean;
    n    pls_integer;
    faculty_rec faculty%rowtype;
begin
    select * into faculty_rec from faculty where faculty = 'ИДиП';
    b1 := sql%found;
    b2 := sql%isopen;
    b3 := sql%notfound;
    n := sql%rowcount;
    dbms_output.put_line(faculty_rec.faculty || ' ' || faculty_rec.faculty_name);
    if   b1 then dbms_output.put_line('b1 = TRUE');
    else      dbms_output.put_line('b1 = FALSE');
    end if;
    if   b2 then dbms_output.put_line('b2 = TRUE');
    else      dbms_output.put_line('b2 = FALSE');
    end if;
    if   b3 then dbms_output.put_line('b3 = TRUE');
    else      dbms_output.put_line('b3 = FALSE');
    end if;
    dbms_output.put_line('n = ' || n);
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



# Неявные курсоры

Атрибут	Описание
%ISOPEN	Всегда FALSE, так как курсор открывается неявно и закрывается сразу после выполнения
%FOUND	NULL перед выполнением. TRUE, если одна или более строк были вставлены, изменены, удалены или одна строка выбрана. Иначе FALSE
%NOTFOUND	NULL перед выполнением. FALSE, если одна или более строк были вставлены, изменены, удалены или одна строка выбрана. Иначе TRUE
%ROWCOUNT	Количество строк, выбранных в курсоре



# Неявные курсоры – UPDATE

```
declare
  b1 boolean;
  b2 boolean;
  b3 boolean;
  n pls_integer;
  auditorium_cur auditorium%rowtype;
begin
  update auditorium set auditorium = '314-1',
    auditorium_name = '314-1',
    auditorium_capacity = 90,
    auditorium_type = 'ЛК'
  where auditorium = '301-1';
  b1 := sql%found;
  b2 := sql%isopen;
  b3 := sql%notfound;
  n := sql%rowcount;
  dbms_output.put_line(auditorium_cur.auditorium_name || ' ' ||
    auditorium_cur.auditorium_capacity|| ' ' ||
    auditorium_cur.auditorium_type);
  if b1 then dbms_output.put_line('b1 = true');
    else dbms_output.put_line('b1 = false');
  end if;
  if b2 then dbms_output.put_line ('b2 = true');
    else dbms_output.put_line('b2 = false');
  end if;
  if b3 then dbms_output.put_line('b3 = true');
    else dbms_output.put_line('b3 = false');
  end if;
  dbms_output.put_line('n = ' || n);
---commit;
rollback;
exception
  when others
    then dbms_output.put_line (sqlerrm);
end;
```

```
bl = true
b2 = false
b3 = false
n = 1
```

# Неявные курсоры – UPDATE

```
declare
b1 boolean;
b2 boolean;
b3 boolean;
n pls_integer;
auditorium_cur auditorium%rowtype;
begin
update auditorium set auditorium = '314-1',
                     auditorium_name = '314-1',
                     auditorium_capacity = 90,
                     auditorium_type = 'ЛК'
    where auditorium = '301-1';
rollback;
b1 := sql%found;
b2 := sql%isopen;
b3 := sql%notfound;
n := sql%rowcount;
dbms_output.put_line(auditorium_cur.auditorium_name || ' ' ||
                     auditorium_cur.auditorium_capacity|| ' ' ||
                     auditorium_cur.auditorium_type);
if b1 then dbms_output.put_line('b1 = true');
      else dbms_output.put_line('b1 = false');
end if;
if b2 then dbms_output.put_line ('b2 = true');
      else dbms_output.put_line('b2 = false');
end if;
if b3 then dbms_output.put_line('b3 = true');
      else dbms_output.put_line('b3 = false');
end if;
dbms_output.put_line('n = ' || n);
---commit;
exception
  when others
    then dbms_output.put_line (sqlerrm);
.
```

```
b1 = false
b2 = false
b3 = true
n = 0
```

# Неявные курсоры – INSERT

```
declare
b1 boolean;
b2 boolean;
b3 boolean;
n pls_integer;
auditorium_cur auditorium%rowtype;
begin
insert into auditorium(auditorium, auditorium_name, auditorium_capacity, auditorium_type)
values ('301-1', '301-1', 90, 'ЛК');
b1 := sql%found;
b2 := sql%isopen;
b3 := sql%notfound;
n := sql%rowcount;
dbms_output.put_line(auditorium_cur.auditorium_name || ' ' ||
                     auditorium_cur.auditorium_capacity|| ' ' ||
                     auditorium_cur.auditorium_type);
if b1 then dbms_output.put_line('b1 = true');
      else dbms_output.put_line('b1 = false');
end if;
if b2 then dbms_output.put_line ('b2 = true');
      else dbms_output.put_line('b2 = false');
end if;
if b3 then dbms_output.put_line('b3 = true');
      else dbms_output.put_line('b3 = false');
end if;
dbms_output.put_line('n = ' || n);
---commit;
rollback;
exception
when others
    then dbms_output.put_line (sqlerrm);
end;
```

```
b1 = true
b2 = false
b3 = false
n = 1
```

# Неявные курсоры – INSERT

```
declare
b1 boolean;
b2 boolean;
b3 boolean;
n pls_integer;
auditorium_cur auditorium%rowtype;
begin
insert into auditorium(auditorium, auditorium_name, auditorium_capacity, auditorium_type)
values ('301-1', '301-1', 90, 'ЛК');
rollback;
b1 := sql%found;
b2 := sql%isopen;
b3 := sql%notfound;
n := sql%rowcount;
dbms_output.put_line(auditorium_cur.auditorium_name || ' ' ||
                     auditorium_cur.auditorium_capacity|| ' ' ||
                     auditorium_cur.auditorium_type);
if b1 then dbms_output.put_line('b1 = true');
      else dbms_output.put_line('b1 = false');
end if;
if b2 then dbms_output.put_line ('b2 = true');
      else dbms_output.put_line('b2 = false');
end if;
if b3 then dbms_output.put_line('b3 = true');
      else dbms_output.put_line('b3 = false');
end if;
dbms_output.put_line('n = ' || n);
---commit;
--rollback;
exception
  when others
    then dbms_output.put_line (sqlerrm);
end;
```

b1 = false  
b2 = false  
b3 = true  
n = 0

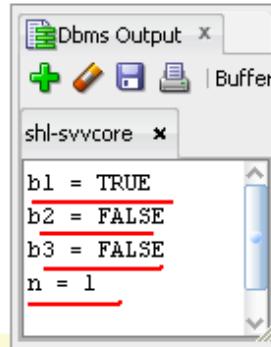
# Неявные курсоры –INSERT

```
declare
b1 boolean;
b2 boolean;
b3 boolean;
n pls_integer;
auditorium_cur auditorium%rowtype;
begin
insert into auditorium(auditorium, auditorium_name, auditorium_capacity, audito
values ('301-1', '301-1', 90, 'ЛК');
b1 := sql%found;
b2 := sql%isopen;
b3 := sql%notfound;
n := sql%rowcount;
dbms_output.put_line(auditorium_cur.auditorium_name || ' ' ||
                     auditorium_cur.auditorium_capacity|| ' ' ||
                     auditorium_cur.auditorium_type);
if b1 then dbms_output.put_line('b1 = true');
      else dbms_output.put_line('b1 = false');
end if;
if b2 then dbms_output.put_line ('b2 = true');
      else dbms_output.put_line('b2 = false');
end if;
if b3 then dbms_output.put_line('b3 = true');
      else dbms_output.put_line('b3 = false');
end if;
dbms_output.put_line('n = ' || n);
---commit;
rollback;
exception
when others
    then dbms_output.put_line (sqlerrm);
end;
```

ORA-00001: нарушено ограничение уникальности (SYSTEM.SYS\_C009869)

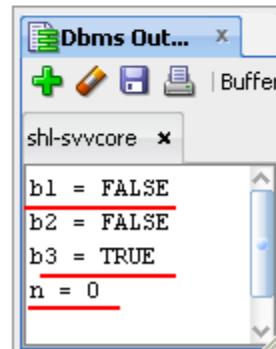
# Неявные курсоры – DELETE

```
-- 11/35.sql
declare
    b1    boolean;
    b2    boolean;
    b3    boolean;
    n    pls_integer;
begin
    delete auditorium where auditorium = '301-4';
    b1 := sql%found;
    b2 := sql%isopen;
    b3 := sql%notfound;
    n := sql%rowcount;
    if   b1 then dbms_output.put_line('b1 = TRUE');
    else      dbms_output.put_line('b1 = FALSE');
    end if;
    if   b2 then dbms_output.put_line('b2 = TRUE');
    else      dbms_output.put_line('b2 = FALSE');
    end if;
    if   b3 then dbms_output.put_line('b3 = TRUE');
    else      dbms_output.put_line('b3 = FALSE');
    end if;
    dbms_output.put_line('n = '||n);
    -- Commit;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



# Неявные курсоры – DELETE

```
-- 11/35.sql
declare
    b1    boolean;
    b2    boolean;
    b3    boolean;
    n     pls_integer;
begin
    delete auditorium where auditorium = 'XXXX';
    b1 := sql%found;
    b2 := sql%isopen;
    b3 := sql%notfound;
    n := sql%rowcount;
    if   b1 then dbms_output.put_line('b1 = TRUE');
    else      dbms_output.put_line('b1 = FALSE');
    end if;
    if   b2 then dbms_output.put_line('b2 = TRUE');
    else      dbms_output.put_line('b2 = FALSE');
    end if;
    if   b3 then dbms_output.put_line('b3 = TRUE');
    else      dbms_output.put_line('b3 = FALSE');
    end if;
    dbms_output.put_line('n = '||n);
    -- commit;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



# Неявные курсоры – RETURNING

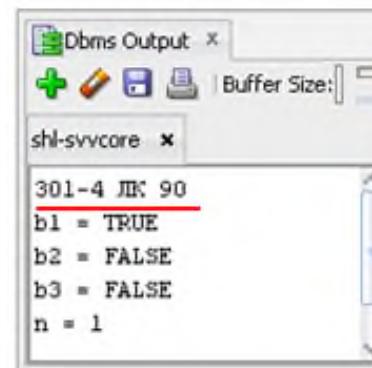
---

- ▶ **RETURNING** в выражениях **INSERT**, **UPDATE**, **DELETE** используется для получения данных, измененных соответствующим выражением
- ▶ Позволяет избежать дополнительного **SELECT** для уточнения результатов



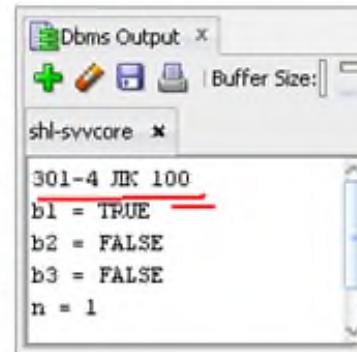
# Неявные курсоры – RETURNING

```
declare
  b1  boolean;
  b2  boolean;
  b3  boolean;
  n   pls_integer;
  a_name      svvcore.auditorium.auditorium name:type;
  a_type      svvcore.auditorium.auditorium type:type;
  a_capacity  svvcore.auditorium.auditorium_capacity:type;
begin
  delete auditorium where auditorium = '301-4'
  returning auditorium name, auditorium type, auditorium capacity
  into a_name, a_type, a_capacity
  dbms_output.put_line(a_name||' '|rtrim(a_type)||'|'||a_capacity);
  b1 := sql%found;
  b2 := sql%isopen;
  b3 := sql%notfound;
  n := sql%rowcount;
  if  b1 then dbms_output.put_line('b1 = TRUE');
  else       dbms_output.put_line('b1 = FALSE');
end if;
  if  b2 then dbms_output.put_line('b2 = TRUE');
  else       dbms_output.put_line('b2 = FALSE');
end if;
  if  b3 then dbms_output.put_line('b3 = TRUE');
  else       dbms_output.put_line('b3 = FALSE');
end if;
  dbms_output.put_line('n = '||n);
  -- commit;
  rollback;
exception
  when others
    then dbms_output.put_line(sqlerrm);
end;
```



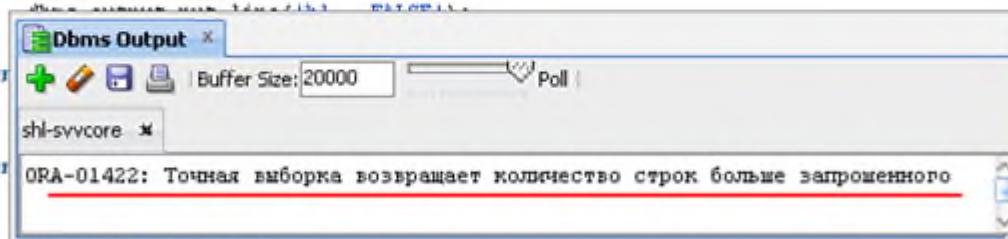
# Неявные курсоры – RETURNING

```
declare
    b1  boolean;
    b2  boolean;
    b3  boolean;
    n   pls_integer;
    a_name      svvcore.auditorium.auditorium_name%type;
    a_type      svvcore.auditorium.auditorium_type%type;
    a_capacity  svvcore.auditorium.auditorium_capacity%type;
begin
    update auditorium set auditorium capacity = 100
    where auditorium = '301-4'
    returning auditorium_name, auditorium_type, auditorium_capacity
    into a_name, a_type, a_capacity :
    dbms_output.put_line(a_name||' '||trim(a_type)||' '||a_capacity);
    b1 := sql%found;
    b2 := sql%isopen;
    b3 := sql%notfound;
    n := sql%rowcount;
    if  b1 then dbms_output.put_line('b1 = TRUE');
    else      dbms_output.put_line('b1 = FALSE');
    end if;
    if  b2 then dbms_output.put_line('b2 = TRUE');
    else      dbms_output.put_line('b2 = FALSE');
    end if;
    if  b3 then dbms_output.put_line('b3 = TRUE');
    else      dbms_output.put_line('b3 = FALSE');
    end if;
    dbms_output.put_line('n = '||n);
    -- commit;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
```



# Неявные курсоры – RETURNING

```
declare
    b1  boolean;
    b2  boolean;
    b3  boolean;
    n   pls_integer;
    a_name      svvcore.auditorium.auditorium_name%type;
    a_type      svvcore.auditorium.auditorium_type%type;
    a_capacity  svvcore.auditorium.auditorium_capacity%type;
begin
    update auditorium set auditorium capacity = 100
    where auditorium type = 'ЛК'
    returning auditorium name, auditorium type, auditorium capacity
    into a_name, a_type, a_capacity ;
    dbms_output.put_line(a_name||' '||rtrim(a_type)||' '||a_capacity);
    b1 := sql%found;
    b2 := sql%isopen;
    b3 := sql%notfound;
    n := sql%rowcount;
    if  b1 then dbms_output.put_line('b1 = TRUE');
    else
        if  b2 then dbms_output.put_line('b2 = TRUE');
        else
            if  b3 then dbms_output.put_line('b3 = TRUE');
            else
                dbms_output.put_line('n = '||n);
                -- commit;
                rollback;
            exception
                when others
                    then dbms_output.put_line(sqlerrm);
            end;
        end if;
    end if;
end;
```



# Явные курсоры

---

- ▶ Этапы открытия явного курсора:
  - ▶ разбор [parse]
  - ▶ связывание переменных [bind]
  - ▶ выполнение [execute]
  - ▶ построение плана выполнения запроса [determining execution plan]
  - ▶ связывание внешних переменных [associating host variables] и курсорных параметров
  - ▶ определение набора данных
  - ▶ выставление указателя текущей строки на первую строку в результирующем наборе данных



# Явные курсоры

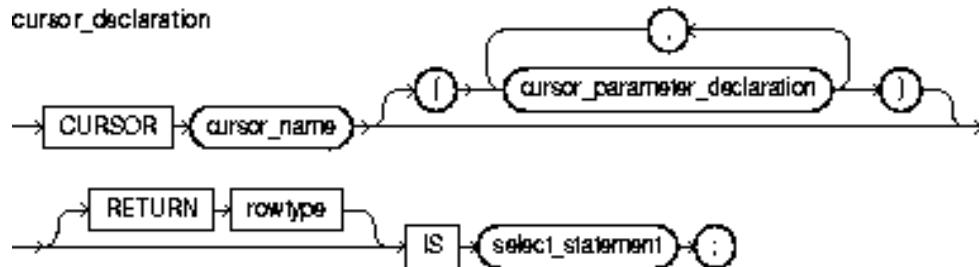
---

- ▶ Открытие явного курсора:
  - ▶ OPEN cursor\_name [(argument [,argument ...])];
- ▶ Выборка из явного курсора:
  - ▶ FETCH cursor\_name INTO record or variable\_list;
- ▶ Закрытие явного курсора:
  - ▶ CLOSE cursor\_name;

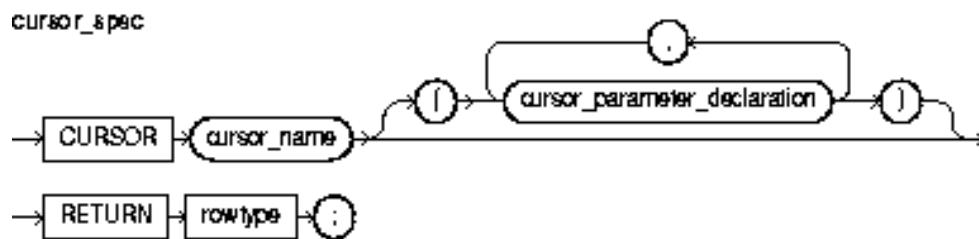


# Курсоры

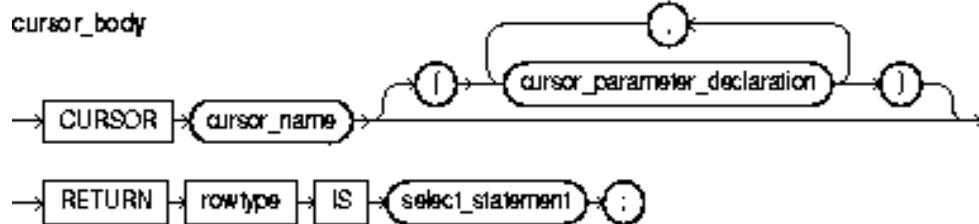
## cursor\_declaration



## cursor\_spec



## cursor\_body



## cursor\_parameter\_declaration



# Явные курсоры

---

- ▶ Объявляются явно в секции декларации блока или в спецификации пакета
- ▶ Объявления явного курсора:
  - ▶ Курсор без параметров
    - ▶ CURSOR company\_cur IS SELECT company\_id FROM company;
  - ▶ Курсор, который принимает параметры через список
    - ▶ CURSOR company\_\_cur (id\_in IN NUMBER) IS
    - ▶ SELECT name FROM company WHERE company\_id = id\_in;
  - ▶ Курсор, который содержит выражение RETURN вместо SELECT
    - ▶ CURSOR company\_cur (id\_in IN NUMBER) RETURN company%ROWTYPE;



# Явные курсоры

```
-- 11/39.sql
declare
    cursor curs_subject is select subject, subject_name, pulpit
                           from subject;
    m_subject      svvcore.subject.subject%type;
    m_subject_name svvcore.subject.subject_name%type;
    m_pulpit       svvcore.subject.pulpit%type;
begin
    open curs_subject;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
    loop
        fetch curs_subject into m_subject, m_subject_name, m_pulpit;
        exit when curs_subject%notfound;
        dbms_output.put_line(''||curs_subject%rowcount||'
                             ||m_subject||'
                             ||m_subject_name||'
                             ||m_pulpit);
    end loop;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
    close curs_subject;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```

```
Dbms Output x
+ - Buffer Size: Poll
shl-svvcore x
rowcount = 0
1 ТП Технология про...
2 УД Управление дан...
3 ИНФ Информатика ИС...
4 КПиЯП Конструировани...
5 ПЗ Представление ...
6 ОПиП Основы педагог...
7 МСОИМ Моделирование ...
8 ПМС Проектирование ...
9 КГиГ Компьютерная г...
10 ЭС Энергосбережени...
11 КМС Компьютерные ...
12 ОТ Охрана труда ...
13 ПП Политология И...
14 ОПП Организация п...
15 ОИ Основы идеоло...
16 КЧИСиТ Классный час ...
17 ФК Физкультура Ф...
18 ПМАПЛ Полиграфическ...
rowcount = 18
```

# Явные курсоры

```
-- 11/40.sql
declare
    cursor curs_subject is select subject, subject_name, pulpits
                           from subject;
    rec_subject svvcore.subject%rowtype;
begin
    open curs_subject;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
    loop
        fetch curs_subject into rec_subject;
        exit when curs_subject%notfound;
        dbms_output.put_line(' '||curs_subject%rowcount||' '
                            ||rec_subject.subject||' '
                            || rec_subject.subject_name||' '
                            || rec_subject.pulpit);
    end loop;
    close curs_subject;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/

```

The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output x'. It contains the following output:

```
rowcount = 0
1 ТП Технология про...
2 УД Управление дан...
3 ИНФ Информатика ИС...
4 КПиИП Конструировани...
5 ПЗ Представление...
6 ОПиИП Основы педагог...
7 МСОИ Моделирование...
8 ПМС Проектирование...
9 КГиГ Компьютерная г...
10 ЭС Энергосбережени...
11 КМС Компьютерные ...
12 ОТ Охрана труда ...
13 ПЛ Политология И...
14 ОИП Организация п...
15 ОИ Основы идеоло...
16 КЧИСиТ Классный час ...
17 ФК Физкультура Ф...
18 ПМАПП Полиграфическ...
rowcount = 18
```

# Явные курсоры

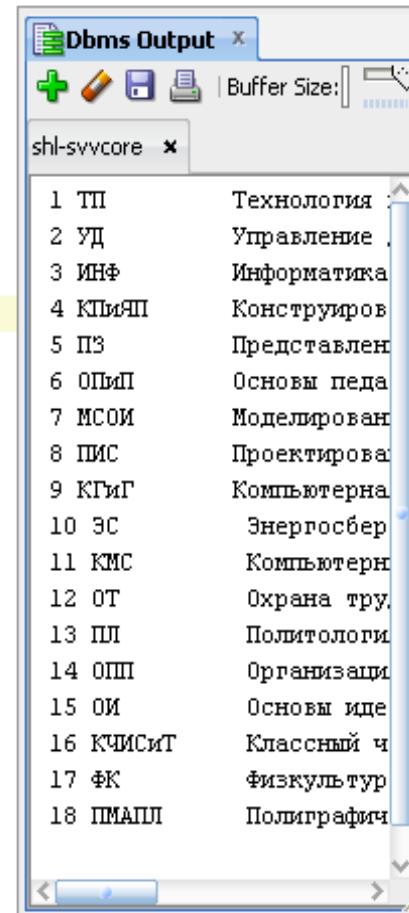
```
-- 11/41.sql
declare
    cursor curs_subject is select subject, subject_name, pulpit
        from subject;
    rec_subject svvcore.subject%rowtype;
begin
    open curs_subject;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
    fetch curs_subject into rec_subject;
    while curs_subject%found
        loop
            dbms_output.put_line(' '||curs_subject%rowcount||' '
                ||rec_subject.subject||' '
                || rec_subject.subject_name||' '
                || rec_subject.pulpit);
            fetch curs_subject into rec_subject;
        end loop;
    dbms_output.put_line('rowcount = '|| curs_subject%rowcount);
    close curs_subject;

exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```

Dbms Output	
	Buffer Size: Poll
shl-svvcore	
rowcount = 0	
1 ТП	Технология прог
2 УД	Управление данны
3 ИНФ	Информатика ИСи
4 КИиП	Конструирование
5 ПЗ	Представление з
6 ОПиП	Основы педагоги
7 МСОИ	Моделирование с
8 ПМС	Проектирование
9 КГиГ	Компьютерная гео
10 ЭС	Энергосбережени
11 КМС	Компьютерные м
12 ОТ	Охрана труда И
13 ПЛ	Политология ИС
14 ОПП	Организация под
15 ОИ	Основы идеолог
16 КЧМСиТ	Классный час ИС
17 ФК	Физкультура ФК
18 ПМДП	Психоматерия

# Явные курсоры

```
-- 11/42.sql
declare
    cursor curs_subject is select subject, subject_name, pulpit
        from subject;
    rec_subject svvcore.subject%rowtype;
begin
    for rec_subject in curs_subject
    loop
        dbms_output.put_line(''||curs_subject%rowcount||'
                            ||rec_subject.subject||'
                            || rec_subject.subject_name||'
                            || rec_subject.pulpit);
    end loop;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. Inside, there is a tab labeled 'shl-svvcore'. The output area displays a list of subjects from the 'subject' table, each row containing a number, subject code, subject name, and pulpit information. The list starts with row 1 (ТП) and ends with row 18 (ПМАП). The output is as follows:

rowcount	subject	subject_name	pulpit
1	ТП	Технология	
2	УД	Управление	
3	ИНФ	Информатика	
4	КИиЯП	Конструиров	
5	ПЗ	Представлен	
6	ОПиП	Основы педа	
7	МСОИ	Моделирован	
8	ПМС	Проектирова	
9	КГиГ	Компьютерна	
10	ЭС	Энергосбер	
11	КМС	Компьютерн	
12	ОТ	Охрана тру	
13	ПП	Политологи	
14	ОПП	Организаци	
15	ОИ	Основы иде	
16	КЧИСиТ	Классный ч	
17	ФК	Физкультур	
18	ПМАП	Полиграфич	

# Явные курсоры с параметрами

```
-- 11/43.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
        is select auditorium, auditorium_capacity
            from auditorium
            where auditorium_capacity >=capacity;
    -- aum  svvcore.auditorium.auditorium%type;
begin
    for aum in curs_auditorium(80)
    loop
        dbms_output.put_line(''||aum.auditorium||' '||aum.auditorium_capacity);
    end loop;

exception
    when others
        then dbms_output.put_line(sqlerrm);

end;
/
```

Dbms Output	
shl-svvcore	
236-1	80
313-1	80
408-2	80
103-4	80
105-4	80
107-4	80
110-4	80
111-4	80
114-4	100
132-4	80
02Б-4	80
229-4	80
314-4	80
320-4	80
429-4	80
?	80
301-4	90

# Явные курсоры с параметрами

```
-- 11/44.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
        is select auditorium, auditorium_capacity
            from auditorium
            where auditorium_capacity >=capacity;
    aum  svvcore.auditorium.auditorium%type;
    cty  svvcore.auditorium.auditorium_capacity%type;
begin
    open curs_auditorium(80);
    fetch curs_auditorium into aum, cty;
    while (curs_auditorium%found)
    loop
        dbms_output.put_line(''||aum||' ||cty);
        fetch curs_auditorium into aum, cty;
    end loop;

exception
    when others
        then dbms_output.put_line(sqlerrm);

end;
/
```

Dbms Output		
Buffer:		
shl-svvcore *		
236-1	80	
313-1	80	
408-2	80	
103-4	80	
105-4	80	
107-4	80	
110-4	80	
111-4	80	
114-4	100	
132-4	80	
02Б-4	80	
229-4	80	
314-4	80	
320-4	80	
429-4	80	
?	80	
301-4	90	

# Явные курсоры с RETURN

```
-- 11/45.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
    return auditorium%rowtype;
    is select * from auditorium
        where auditorium_capacity >=capacity;
    rec_auditorium svvcore.auditorium%rowtype;
begin
    open curs_auditorium(80);
    fetch curs_auditorium into rec_auditorium;
    while (curs_auditorium%found)
    loop
        dbms_output.put_line(' '||rec_auditorium.auditorium
                            ||' '||rec_auditorium.auditorium_capacity);
        fetch curs_auditorium into rec_auditorium;
    end loop;
    close curs_auditorium;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```

Dbms Output		
shl-svvcore		
236-1	80	
313-1	80	
408-2	80	
103-4	80	
105-4	80	
107-4	80	
110-4	80	
111-4	80	
114-4	100	
132-4	80	
02Б-4	80	
229-4	80	
314-4	80	
320-4	80	
429-4	80	
?	80	
301-4	90	

# Явные курсоры

Атрибут	Описание
%ISOPEN	TRUE, если курсор открыт. FALSE, если курсор закрыт.
%FOUND	Ошибка INVALID_CURSOR, если курсор еще не открыт или уже закрыт. NULL перед первой выборкой. TRUE, если запись была успешно выбрана. FALSE, если запись не выбрана.
%NOTFOUND	Ошибка INVALID_CURSOR, если курсор еще не открыт или уже закрыт. NULL перед первой выборкой. FALSE, если запись была успешно выбрана. TRUE, если запись не выбрана.
%ROWCOUNT	Ошибка INVALID_CURSOR, если курсор еще не открыт или уже закрыт. Количество строк, выбранных в курсоре.



# WHERE CURRENT OF

```
-- 12/50.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
        is select auditorium, auditorium_capacity
            from auditorium
            where auditorium_capacity >= capacity for update;
    aum  svvcore.auditorium.auditorium%type;
    cty  svvcore.auditorium.auditorium_capacity%type;
begin
    open curs_auditorium(80);
    fetch curs_auditorium into aum, cty;
    while (curs_auditorium%found)
    loop
        if cty >= 90
        then
            cty := cty * 1.1;
            update auditorium
                set auditorium_capacity = cty
                where current of curs_auditorium;
        end if;
        dbms_output.put_line(''||aum||'||cty);
        fetch curs_auditorium into aum, cty;
    end loop;
    close curs_auditorium;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/

```

Dbms Output x	
	Buffer Size
shl-svvcore x	
236-1	80
313-1	80
408-2	80
103-4	80
105-4	80
107-4	80
110-4	80
111-4	80
114-4	110
132-4	80
02E-4	80
229-4	80
314-4	80
320-4	80
429-4	80
?	80
301-4	99

# WHERE CURRENT OF

```
-- 12/51.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
        is select auditorium, auditorium_capacity
            from auditorium
            where auditorium_capacity >= capacity for update;
    aum  svvcore.auditorium.auditorium%type;
    cty  svvcore.auditorium.auditorium_capacity%type;
begin
    open curs_auditorium(80);
    fetch curs_auditorium into aum, cty;
    while (curs_auditorium%found)
    loop
        if cty >= 90
        then
            delete auditorium
                where current of curs_auditorium;
        end if;
        fetch curs_auditorium into aum, cty;
    end loop;
    close curs_auditorium;

    for pp in curs_auditorium(80)
    loop
        dbms_output.put_line(''||pp.auditorium||
                            '||pp.auditorium_capacity);
    end loop;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
```

auditorium	capacity
236-1	80
313-1	80
408-2	80
103-4	80
105-4	80
107-4	80
110-4	80
111-4	80
132-4	80
02E-4	80
229-4	80
314-4	80
320-4	80
429-4	80
?	80

# Использование ROWID

```
-- 12/52.sql
select rtrim(tlesson), rowid,
       substr(rowid, 1,7) "сегмент",
       substr(rowid,8,3) "№ файла",
       substr(rowid,11,6) "№ блока",
       substr(rowid,16,3) "№ строки"
  from tlesson
  /

```

My Result

SQL | All Rows Fetched: 9 in 0,015 seconds

RTRIM(TLESSON)	ROWID	сегмент	№ файла	№ блока	№ строки
ДП	AAAPkWAAHAAAAdqAAF	AAAPkWA	AHA	AAADqA	AAF
ИП	AAAPkWAAHAAAAdqAAH	AAAPkWA	AHA	AAADqA	AAH
КП	AAAPkWAAHAAAAdqAAE	AAAPkWA	AHA	AAADqA	AAE
КЧ	AAAPkWAAHAAAAdqAAI	AAAPkWA	AHA	AAADqA	AAI
ЛК	AAAPkWAAHAAAAdqAAA	AAAPkWA	AHA	AAADqA	AAA
ЛР	AAAPkWAAHAAAAdqAAB	AAAPkWA	AHA	AAADqA	AAB
ПЗ	AAAPkWAAHAAAAdqAAC	AAAPkWA	AHA	AAADqA	AAC
III	AAAPkWAAHAAAAdqAAG	AAAPkWA	AHA	AAADqA	AAG
ФК	AAAPkWAAHAAAAdqAAD	AAAPkWA	AHA	AAADqA	<b>AAD</b>

# Использование ROWID

```
-- 12/53.sql
declare
    cursor curs_auditorium (capacity svvcore.auditorium.auditorium%type)
        is select auditorium, auditorium_capacity, rowid
            from auditorium
            where auditorium_capacity >= capacity for update;
    sum  svvcore.auditorium.auditorium%type;
    cty  svvcore.auditorium.auditorium_capacity%type;
begin

    for xxx in curs_auditorium(80)
    loop
        case
            when xxx.auditorium_capacity >= 90
                then delete auditorium
                    where rowid = xxx.rowid;
            when xxx.auditorium_capacity >= 80
                then update auditorium
                    set auditorium_capacity = auditorium_capacity+3
                    where rowid = xxx.rowid;
        end case;
    end loop;
    for yyy in curs_auditorium(80)
    loop
        dbms_output.put_line(''||yyy.auditorium||
                            '||yyy.auditorium_capacity);
    end loop;
    rollback;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```

Dbms Output	
shl-svvcore	
236-1	83
313-1	83
408-2	83
103-4	83
105-4	83
107-4	83
110-4	83
111-4	83
132-4	83
02Б-4	83
229-4	83
314-4	83
320-4	83
429-4	83
?	83

# Использование ROWNUM

## ► ROWNUM - логический номер записи в запросе

```
-- 12/54.sql
declare
    cursor curs_auditorium (capacity auditorium.auditorium%type, n number)
        is select auditorium, auditorium_capacity
            from (
                select auditorium, auditorium_capacity
                    from auditorium
                    where auditorium capacity >=capacity
                    order by auditorium_capacity desc
            )
        where rownum <= n;
begin
    for yyy in curs_auditorium(80, 10)
    loop
        dbms_output.put_line(''||yyy.auditorium||
                             '||yyy.auditorium_capacity);
    end loop;
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```

Dbms Output	
shl-svvcore	x
114-4	100
301-4	90
236-1	80
313-1	80
408-2	80
103-4	80
132-4	80
105-4	80
107-4	80
110-4	80

# SELECT FOR UPDATE

---

- ▶ FOR UPDATE используется для блокировки строк в результирующем наборе
- ▶ Использование FOR UPDATE не требует действительного изменения данных
- ▶ Блокировки освобождаются по завершению транзакции
- ▶ Синтаксис:
  - ▶ SELECT ... FROM ... FOR UPDATE [OF column\_reference] [NOWAIT];
  - ▶ NOWAIT – не ожидать освобождения других блокировок



# SELECT FOR UPDATE

```
declare
cursor STUDENT_GRADE_CUR is
select NAMESTUDENT, GRADEDATE, GRADE
from STUDENT inner join STUDENT_GRADES
on STUDENT.STUDENT = STUDENT_GRADES.IDSTUDENT
for update of STUDENT_GRADES.GRADEDATE, STUDENT_GRADES.GRADE;
begin
  null;
end;
```



# Курсорные переменные

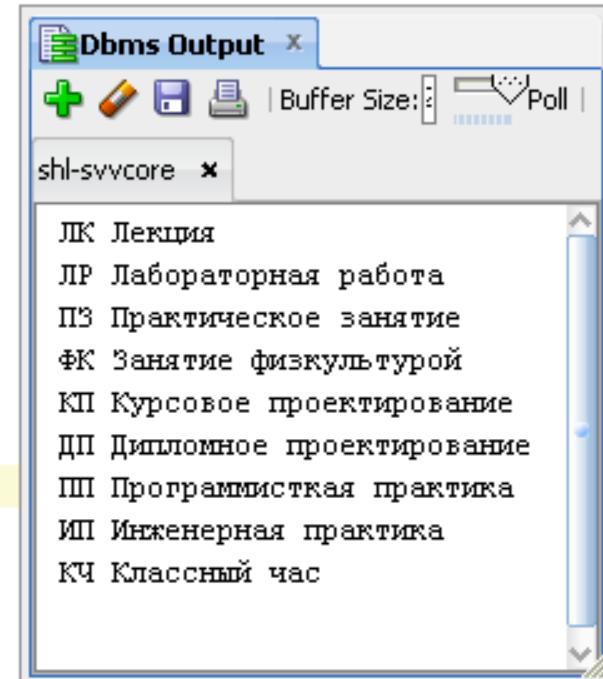
---

- ▶ Курсорные переменные - это структуры данных, которые указывают на курсорный объект
- ▶ Используются для:
  - ▶ Передачи курсора в качестве параметра,
  - ▶ Чтобы отложить связь курсора с SELECT-запросом до выполнения команды OPEN



# Курсорные переменные

```
-- 11/46.sql
declare
  type tlesson type is ref cursor return tlesson%rowtype;
  xcurs tlesson_type;
  rec_tlesson tlesson%rowtype;
begin
  open xcurs for select * from svvcore.tlesson;
  fetch xcurs into rec_tlesson;
  while (xcurs%found)
    loop
      dbms_output.put_line(' '||rtrim(rec_tlesson.tlesson)
                           ||' '||rec_tlesson.tlesson_name);
      fetch xcurs into rec_tlesson;
    end loop;
  close xcurs;
exception
  when others
    then dbms_output.put_line(sqlerrm);
end;
/
```



The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it are icons for new file, edit, save, and print, followed by "Buffer Size:" and a poll button. The main pane is titled "shl-svvcore". It contains a list of course names:

- ЛК Лекция
- ЛР Лабораторная работа
- ПЗ Практическое занятие
- ФК Занятие физкультурой
- КП Курсовое проектирование
- ДП Дипломное проектирование
- ПП Программистская практика
- ИП Инженерная практика
- КЧ Классный час

# Курсорные переменные

---

- ▶ Курсорная переменная, объявленная с помощью **REF CURSOR** без указания **RETURN** может быть связана с любым запросом
- ▶ Курсорная переменная, объявленная с помощью **REF CURSOR** с указанием **RETURN** может быть связана только с запросом, который возвращает результат точно соответствующий числу и типам данных в записи после фразы **RETURN** во время выполнения



# Курсорные переменные

```
SUVCORE@sh1> variable x refcursor;
SUVCORE@sh1> edit
Записано file afiedt.buf

1  declare
2      type tlesson_type is ref cursor return tlesson%rowtype;
3      xcurs tlesson_type;
4      rec_tlesson tlesson%rowtype;
5      begin
6          open xcurs for select * from suvcore.tlesson;
7          :x := xcurs;
8      exception
9          when others
10             then dbms_output.put_line(sqlerrm);
11*     end;
SUVCORE@sh1> /
```

Процедура PL/SQL успешно завершена.

```
SUVCORE@sh1> print x;
```

TLESSON	TLESSON_NAME
ЛК	Лекция
ЛР	Лабораторная работа
ПЗ	Практическое занятие
ФК	Занятие физкультурой
КП	Курсовое проектирование
ДП	Дипломное проектирование
ПП	Программистская практика
ИП	Инженерная практика
КЧ	Классный час

9 строк выбрано.

# Курсорные переменные

```
-- 11/47.041
variable x refcursor;
declare
    type tlesson_type is ref cursor return tlesson%rowtype;
    xcurs tlesson_type;
    rec_tlesson tlesson%rowtype;
begin
    open xcurs for select * from svvcore.tlesson;
    :x := xcurs;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
print x;
```

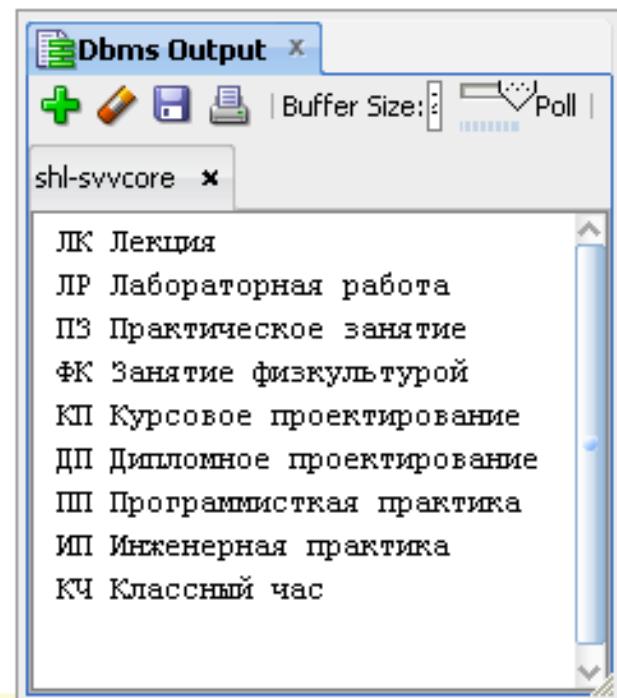
anonymous block completed

X

TLESSON	TLESSON_NAME
ЛК	Лекция
ЛР	Лабораторная работа
ПЗ	Практическое занятие
ФК	Занятие физкультурой
КП	Курсовое проектирование
ДП	Дипломное проектирование
ПП	Программистская практика
ИП	Инженерная практика

# Курсорные переменные

```
declare
    xcurs sys_refcursor;
    rec_tlesson tlesson%rowtype;
begin
    open xcurs for select * from svvcore.tlesson;
    fetch xcurs into rec_tlesson;
    while (xcurs%found)
        loop
            dbms_output.put_line(' '||rtrim(rec_tlesson.tlesson)
                                ||' '||rec_tlesson.tlesson_name);
            fetch xcurs into rec_tlesson;
        end loop;
    close xcurs;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
/
```



The screenshot shows the Oracle SQL Developer interface with the 'Dbms Output' window open. The window title is 'Dbms Output'. It contains a list of course names under the heading 'sh1-svvcore'. The list includes:

- ЛК Лекция
- ЛР Лабораторная работа
- ПЗ Практическое занятие
- ФК Занятие физкультурой
- КП Курсовое проектирование
- ДП Дипломное проектирование
- ПП Программистская практика
- ИП Инженерная практика
- КЧ Классный час

# Курсорные подзапросы

```
declare
    cursor curs aut;
    is select auditorium_type,
        cursor (
            select auditorium
            from auditorium sum
            where aut.auditorium_type = sum.auditorium_type
        )
    from auditorium_type aut;
    curs_sum sys_refcursor;
    aut auditorium_type.auditorium_type%type;
    txt varchar2(1000);
    sum auditorium.auditorium%type;
begin
    open curs_aut;
    fetch curs_aut into aut, curs_sum;
    while (curs_aut%found)
loop
    txt := rtrim(aut)||':';
    loop
        fetch curs_sum into sum;
        exit when curs_sum%notfound;
        txt := txt||','||rtrim(sum);
    end loop;
    dbms_output.put_line(txt);
    fetch curs_aut into aut, curs_sum;
end loop;
close curs_aut;
exception
    when others
        then dbms_output.put_line(sqlerrm);
end;
```

ЛК: ,236-1,313-1,324-1,408-2,103-4,105-4,107-4,110-4,111-4,132-4,02Б-4,229-4,314-4,  
ЛВ-К: ,206-1,301-1,413-1,423-1,304-4  
ЛК-К: ,114-4  
ЛВ-Х:  
ЛВ-СК:

# Динамические курсоры

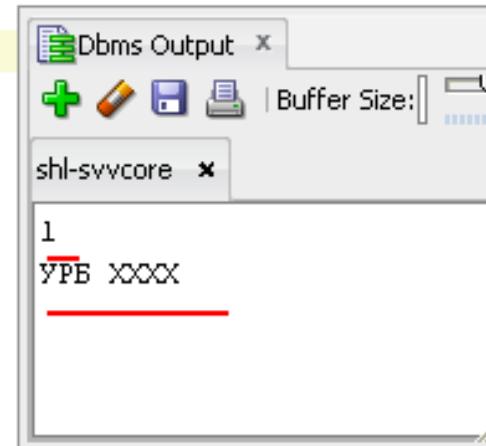
---

- ▶ **EXECUTE IMMEDIATE** - однострочные запросы и DDL команды
- ▶ **OPEN FOR, FETCH** и **CLOSE** - динамические многострочные запросы
- ▶ Синтаксис **EXECUTE IMMEDIATE**:
  - ▶ **EXECUTE IMMEDIATE sql\_statement**
  - ▶ **[INTO {variable [,variable ...] | record}] [USING [IN | OUT | IN OUT] bind\_argument**
  - ▶ **[, [IN | OUT | IN OUT] bind\_argument ...] ] [{RETURNING | RETURN} INTO bind\_argument [,bind\_argument]...];**



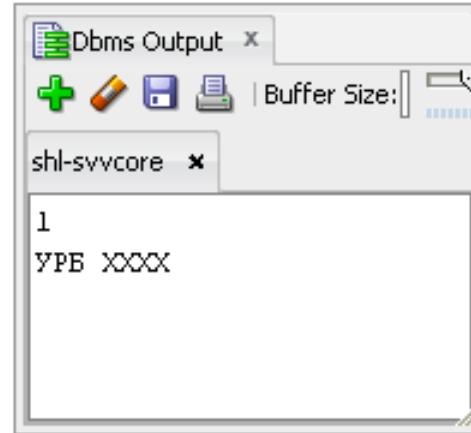
# Динамические курсоры

```
-- 14/08.sql
declare
    stml varchar2(150) := 'update teacher set teacher_name = :1 where teacher = :2';
    t teacher.teacher%type;
    tn teacher.teacher_name%type;
    rec teacher%rowtype;
begin
    t := 'УРБ'; tn := 'XXXX';
    execute immediate stml using tn, t;
    dbms_output.put_line(sql%rowcount);
    select * into rec from teacher where teacher = 'УРБ';
    dbms_output.put_line(rtrim(rec.teacher) || ' ' || rec.teacher_name);
exception
    when others then
        dbms_output.put_line(sqlerrm);
end;
```



# Динамические курсоры

```
-- 14/09.sql
declare
    stml varchar2(150) := 'begin update teacher set teacher_name = :1 where teacher = :2;commit;end';
    t teacher.teacher%type;
    tn teacher.teacher_name%type;
    rec teacher%rowtype;
begin
    t := 'УРБ'; tn := 'XXXX';
    execute immediate stml using tn, t;
    dbms_output.put_line(sql%rowcount);
    select * into rec from teacher where teacher = 'УРБ';
    dbms_output.put_line(rtrim(rec.teacher) || ' ' || rec.teacher_name);
exception
    when others then
        dbms_output.put_line(sqlerrm);
end;
```



# Динамические курсоры

---

- ▶ Для улучшения производительности выполнения SQL выражений можно использовать динамические курсоры со связанными переменными
- ▶ Это позволяет серверу Oracle повторно использовать разобранные SQL выражения из разделяемого пула
  
- ▶ **EXECUTE IMMEDIATE 'INSERT INTO  
dept (deptno, dname, loc) VALUES (:deptno, :dname, :loc)'  
USING deptno\_in, dname\_in, loc\_in;**



# Динамические курсоры - OPEN FOR

```
DECLARE
  TYPE cv_typ IS REF CURSOR;
  cv cv_typ;
  laccount_no NUMBER;
  Ibalance NUMBER;
BEGIN
  OPEN cv FOR
    'SELECT account_no, balance FROM accounts
     WHERE balance < 500';
LOOP
  FETCH cv INTO laccount_no, Ibalance;
  EXIT WHEN cv%NOTFOUND;
  -- Process the row.
END LOOP;
CLOSE cv;
END;
```



# Параметры Oracle, связанные с курсорами

---

- ▶ `cursor_space_for_time` = {TRUE|FALSE} – больший объем памяти для курсоров и никогда не освобождается. Применяется для увеличения скорости работы курсоров при наличии памяти для разделяемого пула.
- ▶ `cursor_sharing` = {EXACT|SIMILAR|FORCE}
- ▶ `open_cursors` - максимальное количество открытых курсоров.
- ▶ `session_cached_cursors` – максимальное количество кэшируемых курсоров для сессии.



# Параметры Oracle, связанные с курсорами

```
SYSTEM@sh1> show parameter cursor;
```

NAME_COL_PLUS_SHOW_PARAM	TYPE	VALUE_COL
<u>cursor_sharing</u>	string	EXACT
<u>cursor_space_for_time</u>	boolean	FALSE
<u>open_cursors</u>	integer	300
<u>session_cached_cursors</u>	integer	20

```
SYSTEM@sh1>
```

```
SYSTEM@sh1> connect svvcore/svvcore@sh1;
```

Соединено.

sh1 - SVVCORE - 23.11.10

```
SVVCORE@sh1> alter session set session_cached_cursors=50;
```

Сеанс изменен.

```
SVVCORE@sh1> alter session set cursor_sharing=similar
```

2 /

Сеанс изменен.

```
SVVCORE@sh1> select * from sys.v$session_cursor_cache;
```

MAXIMUM	COUNT	OPENED_ONCE	OPEN	OPENS	HITS	HIT_RATIO
50	1	1	0	3	3	1
.	.	.	.	.	.	.



# Вопросы?

---



# **ORACLE 12c**

**PL/SQL – Обработка исключений**

**Лекция 11**

# Обработка исключений

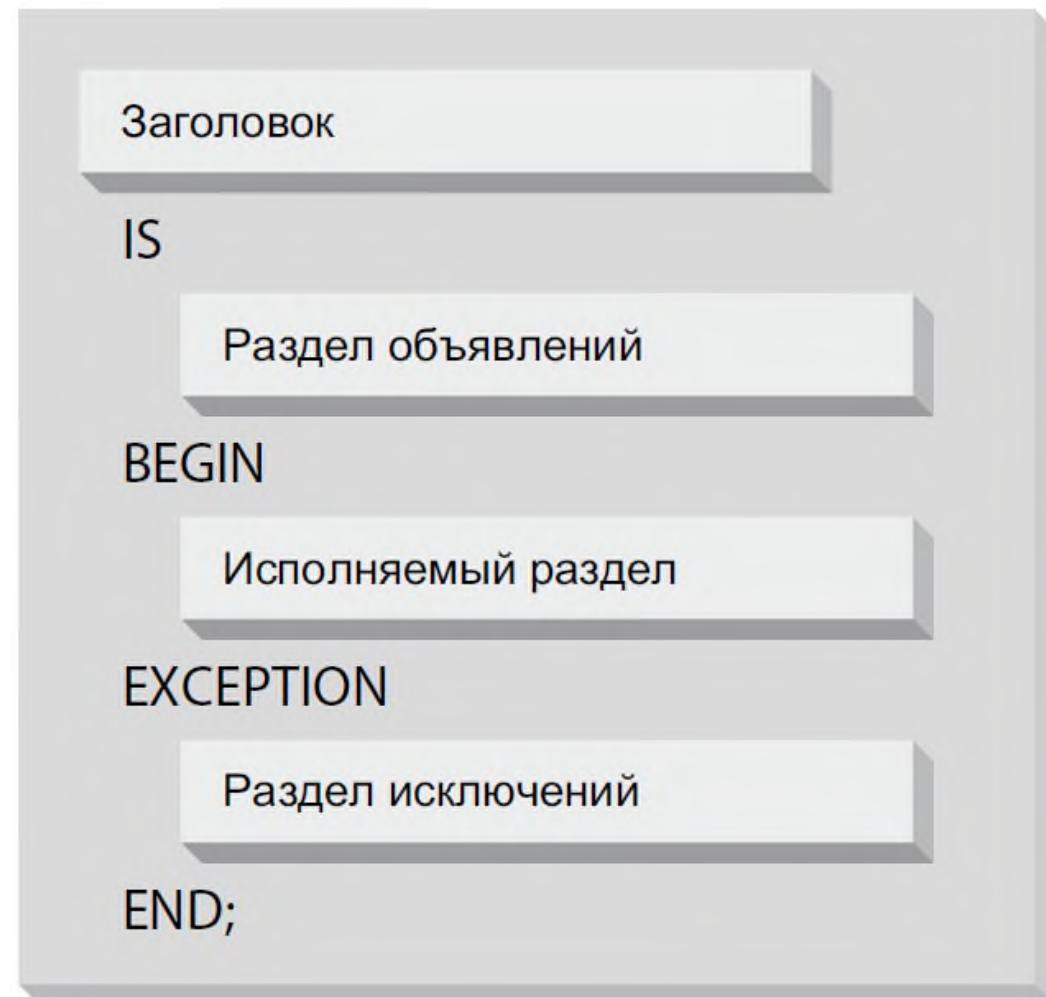
---

- ▶ Исключительная ситуация – событие, возникающее в программе и требующее незамедлительной обработки
- ▶ Два типа исключительных ситуаций:
  - ▶ 1) программно-определяемые исключения;
  - ▶ 2) предопределенные (стандартные) исключения.



# Обработка исключений

- ▶ Ошибка, сгенерированная сервером
- ▶ Ошибка в результате действий пользователя
- ▶ Ошибка, сгенерированная приложением пользователю



# Стратегия обработки исключений

---

- ▶ Как и где будут фиксироваться ошибки, чтобы их можно было просмотреть и откорректировать?
- ▶ Как выдавать пользователю сообщения об ошибках?
- ▶ Нужно ли включать обработку исключений в каждый PL/SQL блок?
- ▶ Как управлять транзакцией в случае ошибки?



# Термины обработки исключений

---

- ▶ Секция исключений – необязательная секция в PL/SQL блоке, которая содержит один или несколько обработчиков исключений
- ▶ RAISE (RAISE\_APPLICATION\_ERROR) – команда, которая прерывает выполнение текущего блока
- ▶ Обработка исключений – перехват ошибки в секции исключений
- ▶ Область действия – часть кода, в рамках которого может быть сгенерировано исключение
- ▶ Распространение исключения – процесс передачи исключений от одного блока другому, если исключение не было обработано



# Термины обработки исключений

---

- ▶ Необработанное исключение – исключение становится необработанным, если оно не обработано блоком самого верхнего уровня
- ▶ Неименованное исключение – исключение, которое имеет код ошибки и сообщение, но не имеет наименования, не может быть использовано в команде RAISE или в секции WHEN
- ▶ Именованное исключение – исключение, которому было определено наименование



# Предопределенные исключения

```
-- 14/05.sql
declare
begin
null;

-- DUP_VAL_ON_INDEX           нарушена уникальность
-- TIMEOUT_ON_RESOURCE        истекло время ожидания ресурса
-- TRANSACTION_BACKED_OUT    запрещенная операция с курсором
-- INVALID_CURSOR             отсутствует подключение к Oracle
-- NOT_LOGGED_ON               данные не найдены
-- NO_DATA_FOUND                не может быть получен ROWID
-- SYS_INVALID_ROWID          не точная выборка в SELECT
-- TOO_MANY_ROWS                 деление на нуль
-- ZERO_DIVIDE                  ошибка USERENV('COMMITSCN')
-- USERENV_COMMITSCN_ERROR     ошибка преобразования в NUMBER
-- INVALID_NUMBER                PL/SQL недостаточно памяти
-- STORAGE_ERROR                 внутренняя ошибка PL/SQL
-- PROGRAMM_ERROR                ошибка преобразование или усечение точности
-- VALUE_ERROR                   переменная PL/SQL и курсорная переменная не совместимы
-- ROWTYPE_MISMATCH              попытка открыть открытый курсор
-- CURSOR_ALREADY_OPEN          попытка присвоить значение атрибуту NULL-объекта
-- ACCESS_INTO_NULL              нет подходящей фразы WHEN в операторе CASE
-- CASE_NOT_FOUND                попытка вызвать метод NULL-объекта
-- SELF_IS_NULL
-- INVALID_PATH
-- INVALID_MODE
-- INVALID_FILEHANDLE
-- INVALID_OPERATION
-- READ_ERROR
-- WRITE_ERROR
-- INTERNAL_ERROR
-- INVALID_MAXLINESIZE
-- INVALID_FILENAME
-- ACCESS_DENIED
-- INVALID_OFFSET
-- DELETE_FAILED
-- RENAME_FAILED
-- NO_DATA_NEEDED
-- collection exceptions
end;
/

```

см. типы исключений для коллекций



# Sqlerrm и sqlcode

The screenshot shows the Oracle SQL Developer interface. On the left, a code editor displays a PL/SQL script named '14/11.sql' containing a loop that prints various SQL error messages using DBMS\_OUTPUT.PUT\_LINE. On the right, the 'Dbms Output' window shows the results of the script execution.

```
-- 14/11.sql
declare
begin
    dbms_output.put_line(sqlerrm(0));
    dbms_output.put_line(sqlerrm(1));
    dbms_output.put_line(sqlerrm(100));
    dbms_output.put_line(sqlerrm(-1));
    dbms_output.put_line(sqlerrm(-6502));
    dbms_output.put_line(sqlerrm(-19999));
    dbms_output.put_line(sqlerrm(-20000));
    dbms_output.put_line(sqlerrm(-20001));
    dbms_output.put_line(sqlerrm(-20800));
    dbms_output.put_line(sqlerrm(-20999));
    dbms_output.put_line(sqlerrm(-21000));
    dbms_output.put_line(sqlerrm(-29001));
    dbms_output.put_line(sqlerrm(-29002));
end;
```

Output from Dbms Output window:

- ORA-0000: normal, successful completion
- User-Defined Exception
- ORA-01403: данные не найдены
- ORA-00001: нарушено ограничение уникальности (.)
- ORA-06502: PL/SQL: ошибка числа или значения
- ORA-19999: была вызвана процедура skip\_row
- ORA-20000: ORA-20000:
- ORA-20001: ORA-20001:
- ORA-20800: ORA-20800:
- ORA-20999: ORA-20999:
- ORA-21000: номер ошибки для raise\_application\_error вне диапазона
- ORA-29001: UNUSED
- ORA-29002: Транспорт SSL обнаружил недопустимый или устаревший сер

# Объявление именованных исключений

---

- ▶ Чтобы обработать исключение, которое не относится к определенным сервером, его необходимо объявить:
  - ▶ `exception_name EXCEPTION;`
- ▶ Имена исключений могут быть использованы только для генерации исключения при помощи `RAISE` и для перехвата исключения в секции `WHEN`



# Связывание исключений с кодом ошибки

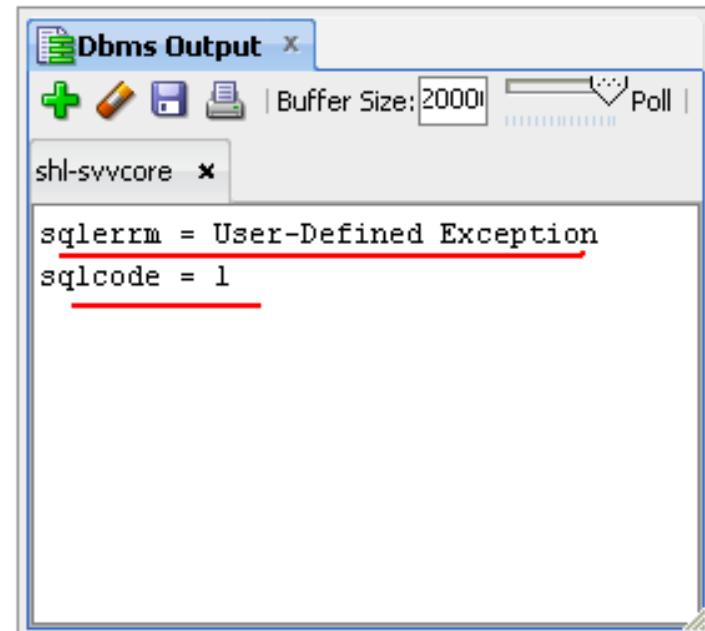
---

- ▶ Синтаксис:
  - ▶ exception\_name EXCEPTION;
  - ▶ PRAGMA EXCEPTION\_INIT (exception\_name, integer);
- ▶ Где exception\_name – наименование исключения, integer – номер(код) ошибки сервера ORACLE, с которым необходимо связать исключение



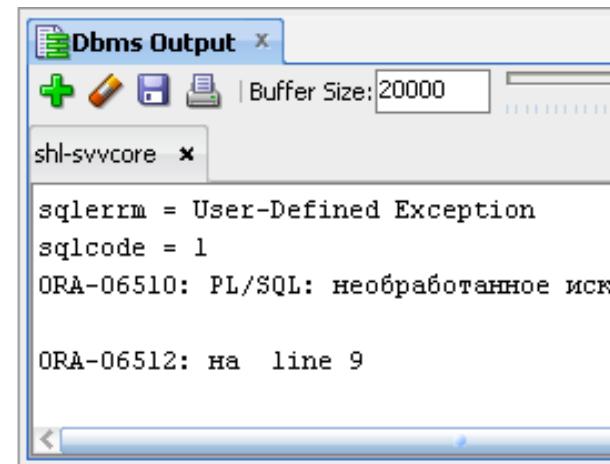
# Генерация и обработка исключений

```
-- 14/Ub.sql
declare
    e_namesake exception;
    n NUMBER(5);
begin
    select count(*) into n
        from teacher t1, teacher t2
       where t1.teacher_name = t2.teacher_name and
             t1.teacher != t2.teacher;
    if n = 0 then raise e_namesake;
end if;
    dbms_output.put_line(n);
exception
    when no_data_found
        then dbms_output.put_line('no_data_found');
    when not_logged_on
        then dbms_output.put_line('no_logged_on');
    when timeout_on_resource
        then dbms_output.put_line('timeout_on_resource');
    when others
        then dbms_output.put_line('sqlerrm = '||sqlerrm);
            dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



# Генерация и обработка исключений

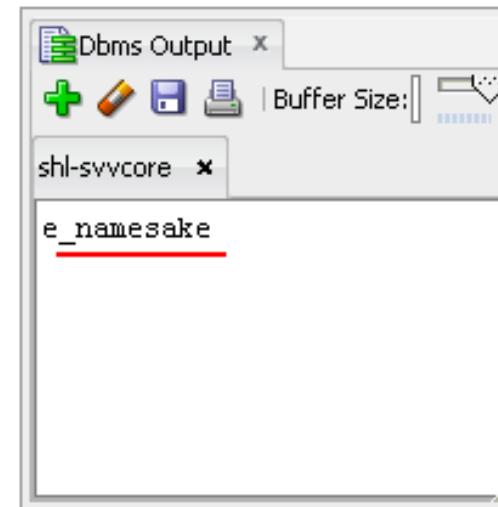
```
-- 14/07.sql
declare
    e_namesake exception;
    n NUMBER(5);
begin
    select count(*) into n
        from teacher t1, teacher t2
       where t1.teacher_name = t2.teacher_name and
             t1.teacher != t2.teacher;
    if n = 0 then raise e_namesake;
    end if;
    dbms_output.put_line(n);
exception
    when no_data_found
        then dbms_output.put_line('no_data_found');
    when not_logged_on
        then dbms_output.put_line('no_logged_on');
    when timeout_on_resource
        then dbms_output.put_line('timeout_on_resource');
    when others
        then dbms_output.put_line('sqlerrm = '||sqlerrm);
        dbms_output.put_line('sqlcode = '||sqlcode);
        dbms_output.put_line(
            substr(dbms_utility.format_error_stack, 1,200)
        );
        dbms_output.put_line(
            substr(dbms_utility.format_error_backtrace, 1,200)
        );
end;
```



# Генерация и обработка исключений

-- 14/08.sql

```
declare
    e_namesake exception;
    n NUMBER(5);
begin
    select count(*) into n
        from teacher t1, teacher t2
       where t1.teacher_name = t2.teacher_name and
             t1.teacher != t2.teacher;
    if n = 0 then raise e_namesake;
    end if;
    dbms_output.put_line(n);
exception
    when no_data_found
        then dbms_output.put_line('no_data_found');
    when not_logged_on
        then dbms_output.put_line('no_logged_on');
    when timeout_on_resource
        then dbms_output.put_line('timeout_on_resource');
    when e_namesake
        then dbms_output.put_line('e_namesake');
    when others
        then dbms_output.put_line('sqlerrm = '||sqlerrm);
                dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



# Генерация и обработка исключений

SQL Worksheet History

-- 14/09.sql

```
1 declare
2     e_namesake exception;
3     pragma exception_init(e_namesake, 100);
4     t teacher.teacher_name%type;
5 begin
6     select tl.teacher into t
7     from teacher tl, teacher t2
8     where tl.teacher_name = t2.teacher_name and
9           tl.teacher != t2.teacher and
10          rownum < 2;
11     dbms_output.put_line(t);
12 exception
13     when no_data_found
14         then dbms_output.put_line('no_data_found');
15     when not_logged_on
16         then dbms_output.put_line('no_logged_on');
17     when timeout_on_resource
18         then dbms_output.put_line('timeout_on_resource');
19     when e_namesake
20         then dbms_output.put_line('e_namesake');
21     when others
22         then dbms_output.put_line('sqlerrm = '||sqlerrm);
23             dbms_output.put_line('sqlcode = '||sqlcode);
24 end;
25
```

Dbms Output x  
shl-svvcore x

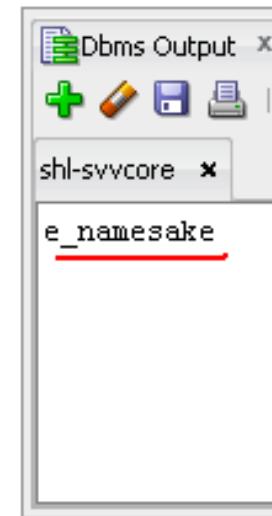
Script Output x  
Task completed in 0,046 seconds

Error report:

ORA-06550: Стока 19, столбец 8:  
PLS-00484: дополнительные исключения 'NO DATA FOUND' и 'E NAMESAKE' должны появляться в том же обработчике исключений  
ORA-06550: Стока 0, столбец 0:  
PLS-00102: неизвестная функция или процедура

# Генерация и обработка исключений

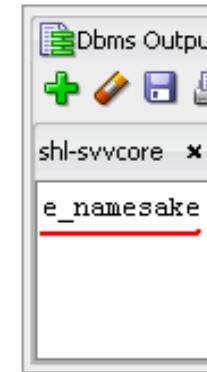
```
-- 14/10.sql
declare
    e_namesake exception;
    pragma exception_init(e_namesake, 100);
    t teacher.teacher_name%type;
begin
    select t1.teacher into t
        from teacher t1, teacher t2
       where t1.teacher_name = t2.teacher_name and
             t1.teacher != t2.teacher and
             rownum < 2;
    dbms_output.put_line(t);
exception
    -- when no_data_found
    --     then dbms_output.put_line('no_data_found');
    when not_logged_on
        then dbms_output.put_line('no_logged_on');
    when timeout_on_resource
        then dbms_output.put_line('timeout_on_resource');
    when e_namesake
        then dbms_output.put_line('e_namesake');
    when others
        then dbms_output.put_line('sqlerrm = '||sqlerrm);
                dbms_output.put_line('sqlcode = '||sqlcode);
end;
```



# RAISE\_APPLICATION\_ERROR

```
-- 14/12.sql
declare
    e_namesake exception;
    pragma exception_init(e_namesake, 100);
    t teacher.teacher_name%type;
begin
    select tl.teacher into t
        from teacher tl, teacher t2
        where tl.teacher_name = t2.teacher_name and
              tl.teacher != t2.teacher and
              rownum < 2;
    dbms_output.put_line(t);
exception
    -- when no_data_found
    when not_logged_on
        then dbms_output.put_line('no_logged_on');
    when timeout_on_resource
        then dbms_output.put_line('timeout_on_resource');
    when e_namesake
        then dbms_output.put_line('e_namesake');
        raise_application_error(-20001,'e_namesake');
    when others
        then dbms_output.put_line('sqlerrm = '||sqlerrm);
        dbms_output.put_line('sqlcode = '||sqlcode);
end;

Error report:
ORA-20001: e namesake
ORA-06512: ha line 20
```



# RAISE\_APPLICATION\_ERROR

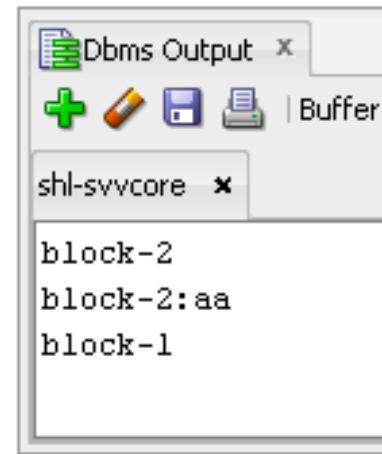
---

- ▶ Определена в пакете **DBMS\_STANDARD**
- ▶ Можно присвоить сообщение об ошибке
- ▶ При выполнении процедуры:
  - ▶ Выполнение блока прерывается
  - ▶ Любые изменения в аргументах **IN OUT** и **OUT** откатываются
  - ▶ Изменения в глобальных структурах (пакетные переменные, объекты базы данных) не откатываются – для отката надо явно выполнить **ROLLBACK**



# Распространение исключения

```
-- 14/13.sql
declare
    aa exception;
begin      -- 1
    begin      -- 2
        dbms_output.put_line('block-2');
        raise aa;
    exception
        when aa  then  dbms_output.put_line('block-2:aa');
    end;      -- 2
    dbms_output.put_line('block-1');
exception
    when others then dbms_output.put_line('block-1:others');
end;      -- 1|
```



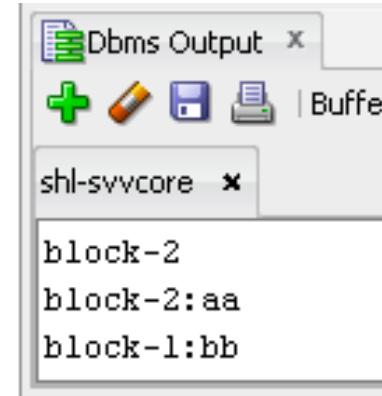
# Распространение исключения

```
-- 14/14.sql
declare
    aa exception;
    bb exception;
begin      -- 1
    begin      -- 2
        dbms_output.put_line('block-2');
        raise bb;
    exception
        when aa then dbms_output.put_line('block-2:aa');
    end;      -- 2
    dbms_output.put_line('block-1');
exception
    when bb      then dbms_output.put_line('block-1:bb');
    when others then dbms_output.put_line('block-1:others');
end;      -- 1
```



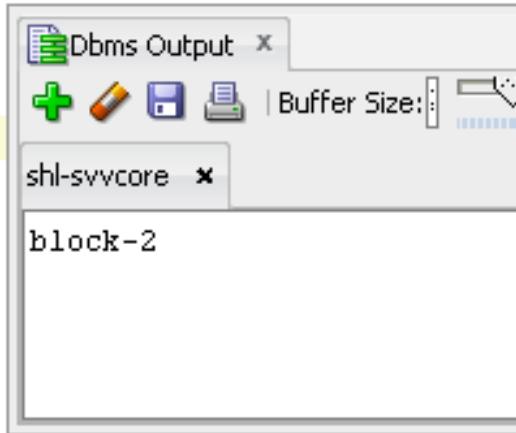
# Распространение исключения

```
17/10/04  
declare  
    aa exception;  
    bb exception;  
    cc exception;  
begin      -- 1  
    begin      -- 2  
        dbms_output.put_line('block-2');  
        raise aa;  
    exception  
        when aa then dbms_output.put_line('block-2:aa');  
                    raise bb;  
        when others then  
            dbms_output.put_line('block-2:others');  
    end;      -- 2  
    dbms_output.put_line('block-1');  
exception  
    when bb then  
        dbms_output.put_line('block-1:bb');  
  
end;      -- 1
```



# Распространение исключения

```
-- 14/16.sql
declare
    aa exception;
    bb exception;
    cc exception;
begin      -- 1
    begin      -- 2
        dbms_output.put_line('block-2');
        raise cc;
    exception
        when aa  then dbms_output.put_line('block-2:aa');
    end;      -- 2
    dbms_output.put_line('block-1');
exception
    when bb      then dbms_output.put_line('block-1:bb');
    -- when others then dbms_output.put_line('block-1:others');
end;      -- 1
```



```
Error report:
ORA-06510: PL/SQL: необработанное исключение, определенное пользователем
ORA-06512: на line 8
ORA-06512: ORA-06512: ORA-06512:
```

# Вопросы?

---



# **ORACLE 12c**

**PL/SQL – встроенные функции**

**Лекция 11**

# Встроенные функции

---

- ▶ Числовые функции
- ▶ Символьные функции
- ▶ Функции по работе с датами
- ▶ Конвертирование
- ▶ Функции обработки ошибок



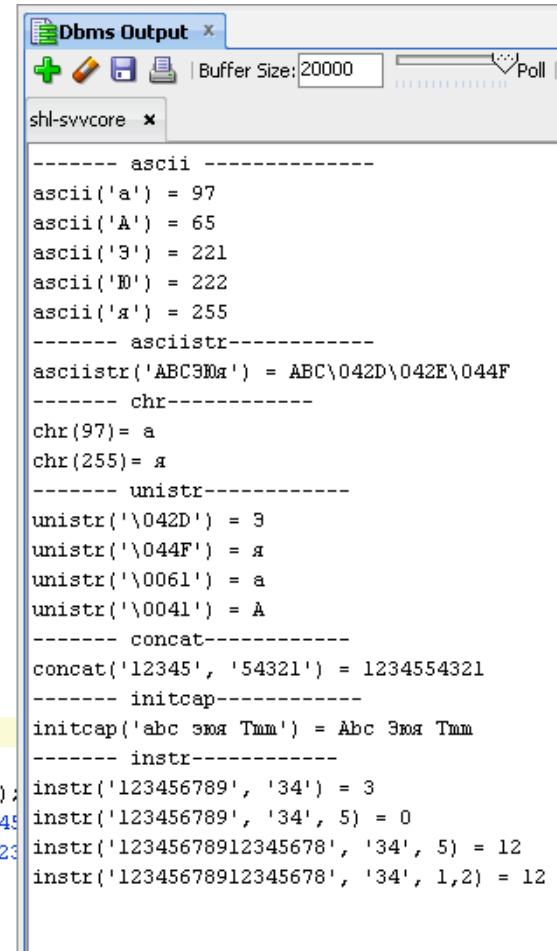
# Числовые функции

```
begin
dbms_output.put_line('abs(-10.493) ='||abs(-10.493));
dbms_output.put_line('ceil(-10.493) ='||ceil(-10.493));
dbms_output.put_line('round(-10.493) ='||round(-10.493));
dbms_output.put_line('round(-10.493,1) ='||round(-10.493,1));
dbms_output.put_line('trunc(-10.493,1) ='||trunc(-10.493,1));
dbms_output.put_line('floor(-10.493) ='||floor(-10.493));
dbms_output.put_line('floor(10.493) ='||floor(10.493));
dbms_output.put_line('remainder(10,3) ='||remainder(10,3)); -- 10%3
dbms_output.put_line('remainder(-10,3) ='||remainder(-10,3)); -- 10%3
dbms_output.put_line('mod(-10,3) ='||mod(-10,3));
dbms_output.put_line('bitand(0,1) ='||to_number(bitand(0,1)));
dbms_output.put_line('bitand(15,7) ='||to_number(bitand(15,7)));
dbms_output.put_line('bitand(5,3) ='||to_number(bitand(5,3)));
-- dbms_output.put_line('width_bucket(21, 0,100,10) ='||WIDTH_BUCKET (21,
dbms_output.put_line('cos(3.14/180*60) ='||cos(3.14/180*60));
dbms_output.put_line('acos(0.5) ='||acos(0.5)/3.14*180);
dbms_output.put_line('sin(3.14/180*60) ='||sin(3.14/180*60));
dbms_output.put_line('asin(0.5) ='||asin(0.5)/3.14*180);
dbms_output.put_line('tan(3.14/180*60) ='||tan(3.14/180*60));
dbms_output.put_line('atan(0.5) ='||atan(0.5)/3.14*180);
dbms_output.put_line('exp(1) ='||exp(1));
dbms_output.put_line('exp(0) ='||exp(0));
dbms_output.put_line('power(5,2) ='||power(5,2));
dbms_output.put_line('power(25,1/2) ='||power(25,1/2));
dbms_output.put_line('power(5,-2) ='||power(5,-2));
dbms_output.put_line('sqrt(16) ='||sqrt(16));
dbms_output.put_line('log(100,10) ='||log(100,10));
dbms_output.put_line('log(10,100) ='||log(10,100));
dbms_output.put_line('log(2,16) ='||log(2,16));
dbms_output.put_line('sign(-25.7) ='||sign(-25.7));
dbms_output.put_line('sign(25.7) ='||sign(25.7));
dbms_output.put_line('sign(0) ='||sign(0));
dbms_output.put_line('greatest(-1,3,45,9,1) ='||greatest(-1,3,45,9,1));
```

```
abs(-10.493) = 10,493
ceil(-10.493) = -10
round(-10.493) = -10
round(-10.493,1) = -10,5
trunc(-10.493,1) = -10,4
floor(-10.493) = -11
floor(10.493) = 10
remainder(10,3) = 1
remainder(-10,3) = -1
mod(-10,3) = -1
bitand(0,1) = 0
bitand(15,7) = 7
bitand(5,3) = 1
cos(3.14/180*60) = ,5004596890082057
acos(0.5) = 60,030432871142545957884
sin(3.14/180*60) = ,8657598394923444
asin(0.5) = 30,015216435571272978942
tan(3.14/180*60) = 1,729929220089790
atan(0.5) = 26,578525356734108572791
exp(1) = 2,7182818284590452353602874
exp(0) = 1
power(5,2) = 25
power(25,1/2) = 5,0000000000000000000000000000000
power(5,-2) = ,04
sqrt(16) = 4
log(100,10) = ,5
log(10,100) = 2
log(2,16) = 3,9999999999999999999999999999999
sign(-25.7) = -1
sign(25.7) = 1
sign(0) = 0
greatest(-1,3,45,9,1) = 45
```

# Символьные функции

```
-- 12/55.sql
declare
  vvv varchar(200);
begin
dbms_output.put_line('----- ascii -----');
  dbms_output.put_line('ascii(''a'') = ''||ascii(''a''));
  dbms_output.put_line('ascii(''A'') = ''||ascii(''A''));
  dbms_output.put_line('ascii(''3'') = ''||ascii(''3''));
  dbms_output.put_line('ascii(''W'') = ''||ascii(''W''));
  dbms_output.put_line('ascii(''я'') = ''||ascii(''я''));
dbms_output.put_line('----- asciistr-----');
  dbms_output.put_line('asciistr(''ABCЭMa'') = ''||asciistr(''ABCЭMa''));
dbms output.put line('----- chr-----');
  dbms_output.put_line( 'chr(97)= ''||chr(97));
  dbms_output.put_line( 'chr(255)= ''||chr(255));
dbms output.put line('----- unistr-----');
  dbms_output.put_line( 'unistr(''\042D'') = ''||unistr(''\042D''));
  dbms_output.put_line( 'unistr(''\044F'') = ''||unistr(''\044F''));
  dbms_output.put_line( 'unistr(''\0061'') = ''||unistr(''\0061''));
  dbms_output.put_line( 'unistr(''\0041'') = ''||unistr(''\0041''));
dbms_output.put_line('----- concat-----');
  dbms_output.put_line( 'concat(''12345'', ''54321'') = ''||concat(''12345'', ''54321''));
dbms output.put line('----- initcap-----');
  dbms_output.put_line( 'initcap(''abc эмя Тmm'') = ''||initcap(''abc эмя Тmm''));
dbms output.put line('----- instr-----');
  dbms_output.put_line( 'instr(''123456789'', ''34'') = ''||instr(''123456789'', ''34''));
  dbms_output.put_line( 'instr(''123456789'', ''34'', 5) = ''||instr(''123456789'', ''34'', 5));
  dbms_output.put_line( 'instr(''12345678912345678'', ''34'', 5) = ''||instr(''12345678912345678'', ''34'', 5));
  dbms_output.put_line( 'instr(''12345678912345678'', ''34'', 1,2) = ''||instr(''12345678912345678'', ''34'', 1,2))
```



The screenshot shows the Oracle SQL Developer Dbms Output window with the following content:

```
Dbms Output x
+ - Buffer Size: 20000 Poll
shl-svvcore x
----- ascii -----
ascii('a') = 97
ascii('A') = 65
ascii('3') = 221
ascii('W') = 222
ascii('я') = 255
----- asciistr-----
asciistr('ABCЭMa') = ABC\042D\042E\044F
----- chr-----
chr(97)= а
chr(255)= я
----- unistr-----
unistr(''\042D'') = ю
unistr(''\044F'') = я
unistr(''\0061'') = а
unistr(''\0041'') = А
----- concat-----
concat(''12345'', ''54321'') = 1234554321
----- initcap-----
initcap(''abc эмя Тmm'') = Аbc Эмя Тmm
----- instr-----
instr(''123456789'', ''34'') = 3
instr(''123456789'', ''34'', 5) = 0
instr(''12345678912345678'', ''34'', 5) = 12
instr(''12345678912345678'', ''34'', 1,2) = 12
```

# Работа с датами

```
declare
    v varchar2(50);
begin
    dbms_output.put_line('current_date = '|| current_date);
    dbms_output.put_line('current_timestamp = '|| current_timestamp);
    dbms_output.put_line(sysdate);
    dbms_output.put_line(localtimestamp);
    dbms_output.put_line(sys_extract_utc(timestamp '2000-03-28 11:30:00.00 -08:00'));
    dbms_output.put_line(next_day('01-12-10', 'суббота'));
    dbms_output.put_line(last_day(to_date('2003/03/15', 'yyyy/mm/dd')));
    dbms_output.put_line(last_day(to_date('2003/02/03', 'yyyy/mm/dd')));
    dbms_output.put_line(last_day(to_date('2004/02/03', 'yyyy/mm/dd')));
    dbms_output.put_line(dbtimezone); -- CREATE/ALTER DATABASE
    dbms_output.put_line(sessiontimezone); -- CREATE/ALTER SESSION
    dbms_output.put_line(tz_offset('Europe/Minsk'));
    dbms_output.put_line(extract(year from date '2003-08-22'));
    dbms_output.put_line(extract(month from date '2003-08-22'));
    dbms_output.put_line(extract(day from date '2003-08-22'));
    dbms_output.put_line(months_between(sysdate, sysdate+100));
    dbms_output.put_line(months_between(sysdate+100, sysdate));
    dbms_output.put_line(round(to_date ('01-12-10'),'YEAR'));
    dbms_output.put_line(round(to_date ('02-12-10'),'MONTH'));
    dbms_output.put_line(round(to_date ('02-12-10'),'DAY'));
    dbms_output.put_line(round(to_date ('02-12-10'),'Q'));
    dbms_output.put_line(trunc(to_date ('01-12-10'),'YEAR'));
    dbms_output.put_line(trunc(to_date ('02-12-10'),'Q'));
    dbms_output.put_line(new_time (to_date ('2003/11/01 01:45', 'yyyy/mm/dd HH24:MI'), 'AST', 'MST'));
```

The screenshot shows the Oracle SQL Developer interface with a code editor on the left and an output window on the right. The output window title is 'shl-svvcore \*'. It displays the results of the PL/SQL code execution, which includes:

- current\_date = 01.12.10
- current\_timestamp = 01-ДЕК-10 11.09.59,10
- 01.12.10
- 01-ДЕК-10 11.09.59,109000000 PM
- 28-МАР-00 07.30.00,0000000000 PM
- 04.12.10
- 31.03.03
- 28.02.03
- 29.02.04
- +00:00
- Europe/Minsk
- +02:00
- 2003
- 8
- 22
- 3,32258064516129032258064516129032258065
- 3,32258064516129032258064516129032258065
- 01.01.11
- 01.12.10
- 29.11.10
- 01.01.11
- 01.01.10
- 01.10.10
- 31.10.03

# Функции конвертирования

```
-- 13/03.sql
declare
    v varchar2(3):='A';
begin
    dbms_output.put_line(convert('АВВГДЕ','WE8ISO8859P1'));
    dbms_output.put_line(to_number('1210.73', '9999.99'));
    dbms_output.put_line(to_number('546', '999'));
    dbms_output.put_line(to_number('23', '99'));
    dbms_output.put_line(to_char(23, '99'));
    dbms_output.put_line(to_char(1210.73777, '9999.99'));
    dbms_output.put_line(to_date('01.12.2010', 'DD.MM.YYYY'));
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```

The screenshot shows the Oracle SQL Developer interface with two panes. The left pane contains the PL/SQL code from the previous block. The right pane, titled 'Dbms Output', displays the results of the code execution:

Output
iiiii
1210,73
546
23
23
1210.74
01.12.10

# Sqlerrm и sqlcode

---

- ▶ Функция SQLERRM возвращает сообщение об ошибке, связанной с исключительной ситуацией
- ▶ Функция SQLCODE возвращает номер ошибки, связанной с исключительной ситуацией



# Функции регулярных выражений

---

- ▶ Регулярные выражения - формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов



# Функции регулярных выражений

---

- ▶ REGEXP\_LIKE выбирает все строки, соответствующие заданному шаблону
  - ▶ REGEXP\_LIKE (source\_char, pattern [, match\_parameter ] )
- ▶ REGEXP\_INSTR определяет местоположение вхождения шаблона в строку
  - ▶ REGEXP\_INSTR(source\_char, pattern [, start\_position [, nth\_appearance [, return\_option [, match\_parameter [, sub\_expression ] ] ] ] ] )



# Функции регулярных выражений

---

- ▶ REGEXP\_REPLACE заменяет шаблон выражения на заданный
  - ▶ REGEXP\_REPLACE(source\_char, pattern [, replacement\_string [, start\_position [, nth\_appearance [, match\_parameter ] ] ] ] )
- ▶ REGEXP\_SUBSTR выделяет из строки шаблон
  - ▶ REGEXP\_SUBSTR(source\_char, pattern [, start\_position [, nth\_appearance [, match\_parameter [, sub\_expression ] ] ] ] )
- ▶ REGEXP\_COUNT определяет количество вхождений
  - ▶ REGEXP\_COUNT (source\_char, pattern [, position [, match\_param]])

# Метасимволы привязки

Мета-символ	Описание	Пример
^	Привязать выражение к началу строки	REGEXP_LIKE(str,'^t') Результат: test11 => true 11123345 => false
\$	Привязать выражение к концу строки	REGEXP_LIKE(str,'\$5') Результат: test11 => false 11123345 => true



# Квантификаторы и операторы повтора

Квантификат ор	Описание	Пример
*	Встречается 0 и более раз	<code>REGEXP_REPLACE(str, '11*', '1')</code> Результат: <code>test11 =&gt; test1</code> <code>11123345 =&gt; 123345</code>
?	Встречается 0 или 1 раз	
+	Встречается 1 и более раз	<code>REGEXP_LIKE(str,'5+')</code> Результат: <code>test11 =&gt; false</code> <code>11123345 =&gt; true</code>
{m}	Встречается ровно m раз	<code>REGEXP_LIKE(str,'3{2}')</code> Результат: <code>test11 =&gt; false</code> <code>11123345 =&gt; true</code>
{m,}	Встречается по крайней мере m раз	
{m, n}	Встречается по крайней мере m раз, но не более n раз	

# Предопределенные символьные классы POSIX

Класс символов	Описание
.	Любой символ
[:alpha:]	Буквы
[:lower:]	Буквы в нижнем регистре
[:upper:]	Буквы в верхнем регистре
[:digit:]	Цифры
[:alnum:]	Буквы и цифры
[:space:]	Пробелы (не печатаемые символы), такие как перевод каретки, новая строка, вертикальная табуляция и подача страницы
[:punct:]	Знаки препинания
[:cntrl:]	Управляющие символы (не печатаемые)
[:print:]	Печатаемые символы



# Альтернативное сопоставление и группировка

Метасимвол	Описание	
	Альтернатива	Разделяет альтернативные варианты, часто используется с оператором группировки ()  SELECT first_name, last_name FROM employees WHERE REGEXP_LIKE (first_name, '^Ste(v ph)en\$');
( )	Группа	Группирует выражения для альтернативы  Steven King Steven Markle Stephen Stiles
[char]	Список символов	Список символов, которые должны присутствовать в строке  SELECT last_name FROM employees WHERE REGEXP_LIKE (last_name, '([aeiou])\I', 'i');
[^char]	Список символов	Список символов, которые не должны присутствовать в строке  De Haan Greenberg Khoo Gee Greene Lee Bloom Feeney

# Метасимвол ссылки

Метасимвол	Описание
\digit	<p>Обратная косая черта</p> <p>Предыдущее сопоставление с соответствующим номером выражения в скобках</p> <p>Обратная косая черта может иметь другое значение</p>



# Вопросы?

---



# **ORACLE 12c**

**PL/SQL Программные модули**

**Лекция 12**

# Программные модули

---

- ▶ Локальные
- ▶ Хранимые



# Локальные программные модули

---

- ▶ Локальный программный модуль – это процедура или функция, определенная в секции декларации PL/SQL блока
- ▶ Объявление локальных процедур и функций должно размещаться в конце секции декларации после всех типов, записей, курсоров, переменных и исключений
- ▶ Локальные процедуры и функции могут быть использованы только в рамках блока, в котором они объявлены
- ▶ Локальные процедуры и функции могут быть перегружены



# Перегрузка программных модулей

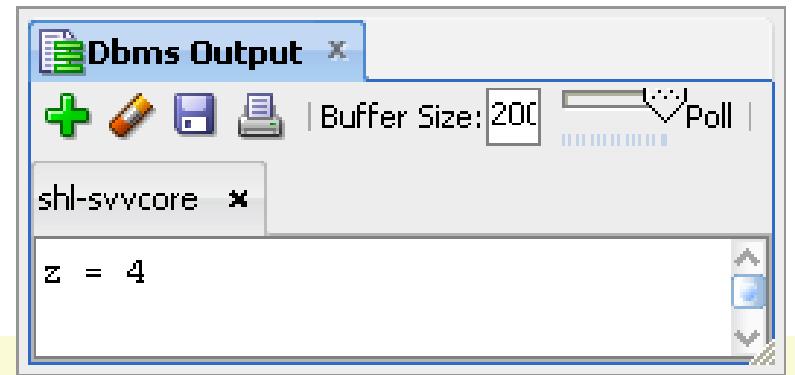
---

- ▶ Параметры должны отличаться семейством (number, character, datetime, boolean)
- ▶ Тип программного модуля должен отличаться – можно перегружать процедуру и функцию с одинаковым именем и списком параметров
- ▶ Число параметров должно быть разным



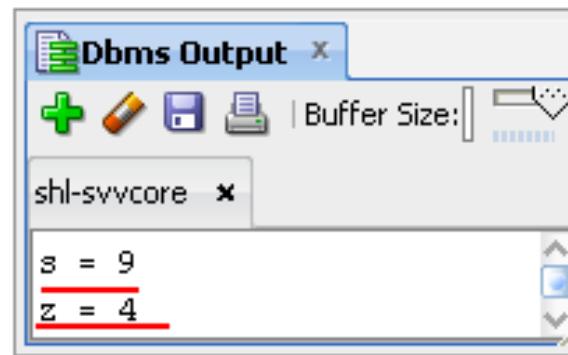
# Локальные процедуры

```
-- 13/08.sql
declare
    x number(3):= 4;
    y number(3):= 5;
    z number(3);
    procedure summod5 (x1 number, x2 number, x3 out number)
    is
        z number(3):= 5;
    begin
        x3 := mod(x1+ x2,z);
    end summod5;
begin
    summod5(x,y,z);
    dbms_output.put_line('z = '||z);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



# Локальные функции

```
-- 13/09.sql
declare
    x number(3):= 4;
    y number(3):= 5;
    z number(3);
    s number(5);
    function summod5 (x1 number, x2 number, x3 out number)
        return number is
            z number(3):= 5;
    begin
        x3 := mod(x1+ x2,z);
        return (x1+x2);
    end summod5;
begin
    s := summod5(x,y,z);
    dbms_output.put_line('s = '||s);
    dbms_output.put_line('z = '||z);
exception
    when others then dbms_output.put_line(sqlerrm);
end;
/
```



# Программные модули

---

- ▶ Процедура
- ▶ Функция
- ▶ Пакет
- ▶ Триггер
- ▶ Объектный тип
- ▶ Хранимые процедуры на Java



# Процедура

---

- ▶ Процедура – именованный модуль, который выполняет одно или несколько выражений и может принимать или возвращать значения через список параметров



# Привилегии

---

- ▶ Для создания процедур необходима привилегия  
`create procedure`

```
SUVCORE@sh1>
SUVCORE@sh1> connect system/system@sh1;
Соединено.
sh1 - SYSTEM - 12.12.10
SYSTEM@sh1> grant create procedure to RLSUV;
Привилегии предоставлены.
```

---

```
SYSTEM@sh1>
SYSTEM@sh1>
```

# Параметры

---

- ▶ Наименование
- ▶ Тип данных
- ▶ Режим передачи
- ▶ Начальное значение



## Тип данных параметров

---

- ▶ PL/SQL или программно-определенный
- ▶ Не может быть ограничен по размеру
- ▶ Размер определяется через вызывающую программу или через связанное объявление переменной



# Параметры

---

- ▶ Типы параметров:
  - ▶ IN
  - ▶ OUT
  - ▶ IN OUT
- ▶ При выполнении:
  - ▶ Значения OUT устанавливаются в NULL
  - ▶ Значения IN OUT остаются неизменными
  - ▶ При ошибке присвоения для параметров откатываются, кроме NOCOPY



# Значения по умолчанию

---

- ▶ IN, IN OUT
- ▶ Можно не задавать при вызове



# Передача параметров

---

- ▶ Позиционный – каждое значение в списке аргументов вызова ставится в соответствие формальному параметру по порядку.

*Emplid\_to\_name(23, name, surname);*

- ▶ Именованный – явно связывает аргументы при вызове с параметрами по именам.

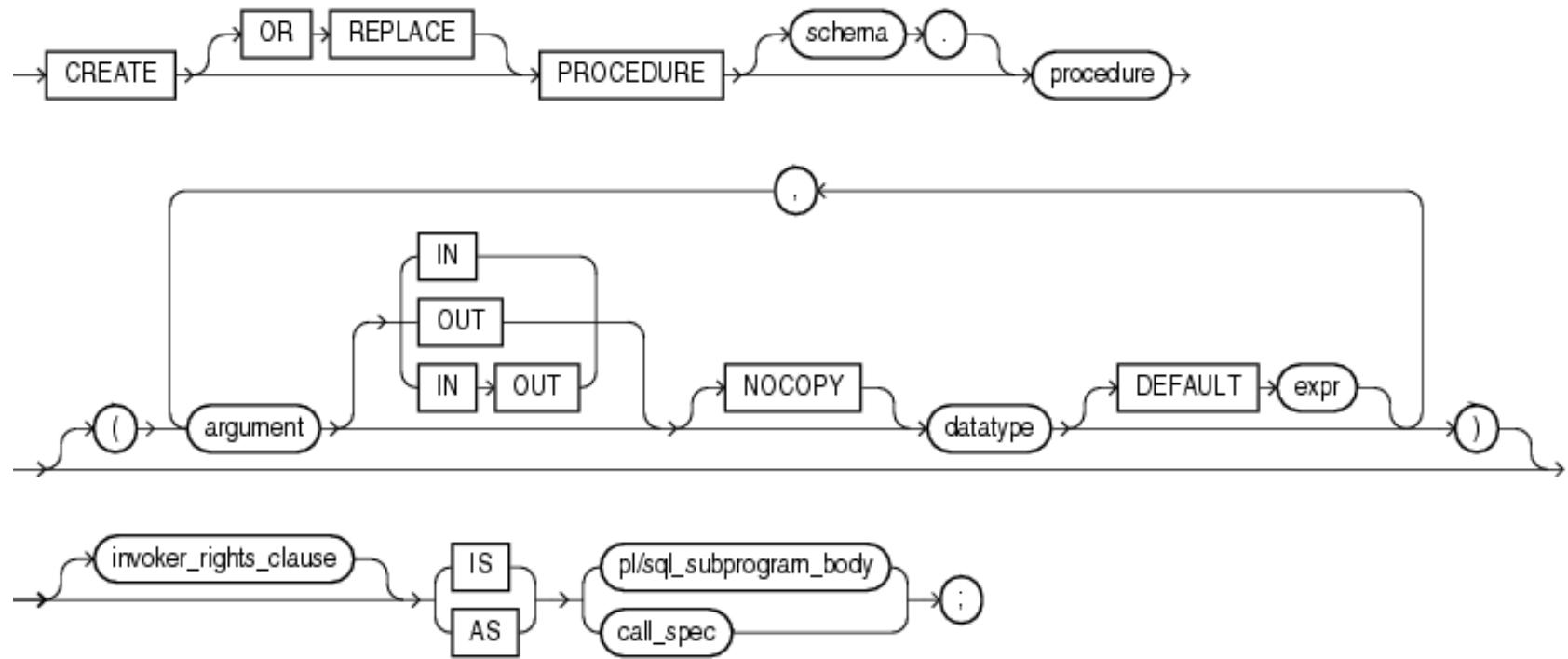
*Emplid\_to\_name(in\_id =>23, out\_name=> name, out\_surname =>surname);*

- ▶ Можно комбинировать оба метода, пока позиционные аргументы стоят слева.

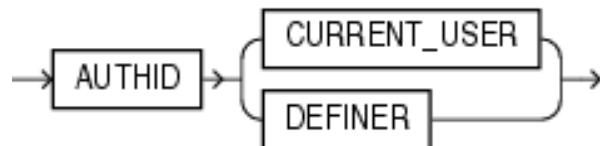
▶ *Emplid\_to\_name(23, name, out\_surname =>surname);*



# Синтаксис



`invoker_rights_clause`



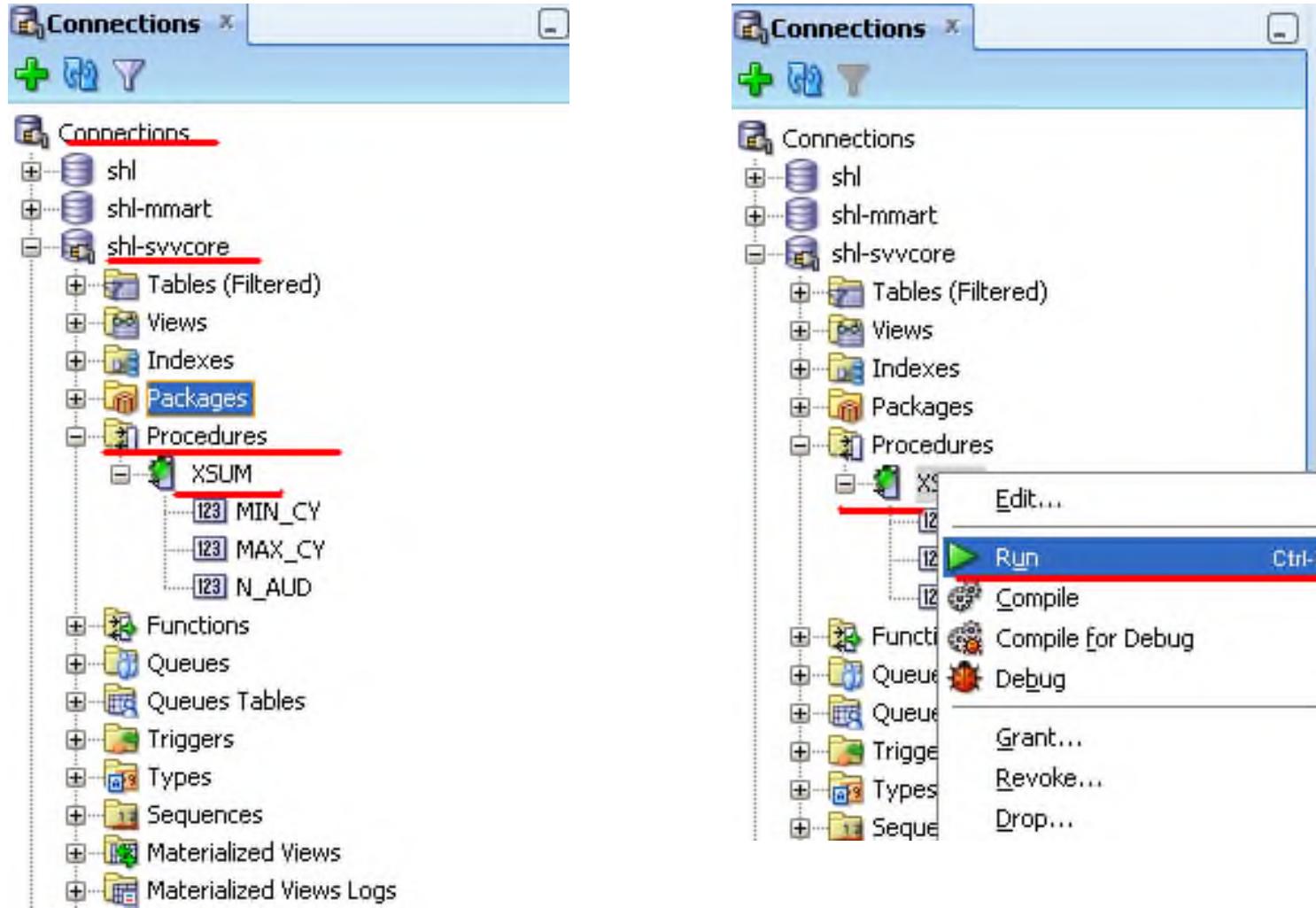
# Процедуры

```
-- 16/01.sql
--select * from dba_sys_privs where grantee = 'RLSVV';
-- select * from user_role_privs where username = 'SVVCORE';
-- grant create procedure to RLSVV

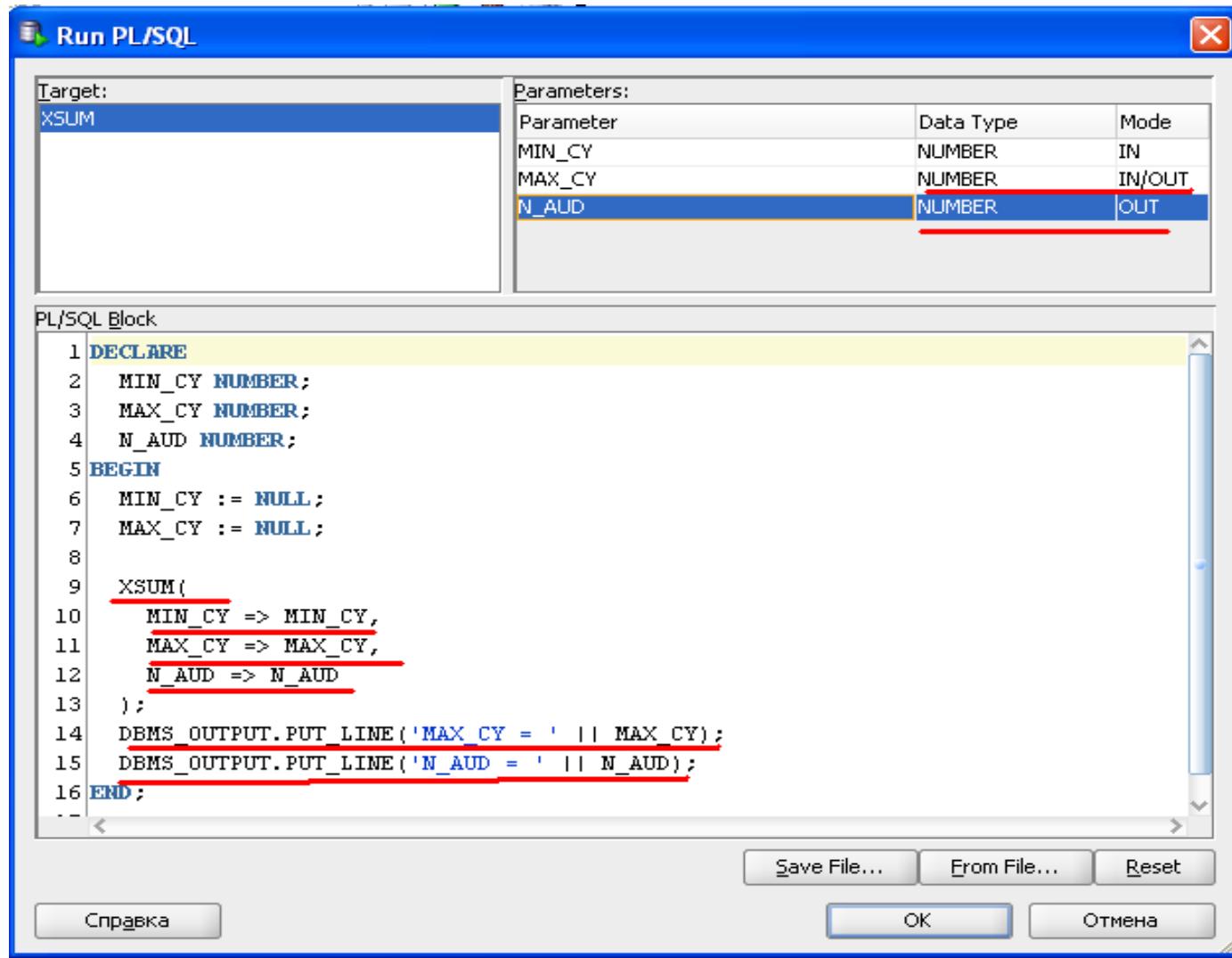
create or replace procedure svvcore.xsum(
    min_cy    in      svvcore.auditorium.auditorium_capacity%type,
    max_cy    in out   svvcore.auditorium.auditorium_capacity%type,
    n_aud     out      number
)
is
    m_max_cy  svvcore.auditorium.auditorium_capacity%type;
    m_n_aud   number := 0;
begin
    select count(*), max(auditorium_capacity) into m_n_aud, m_max_cy from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
exception
    when others then dbms_output.put_line(sqlerrm);
end xsum;
```



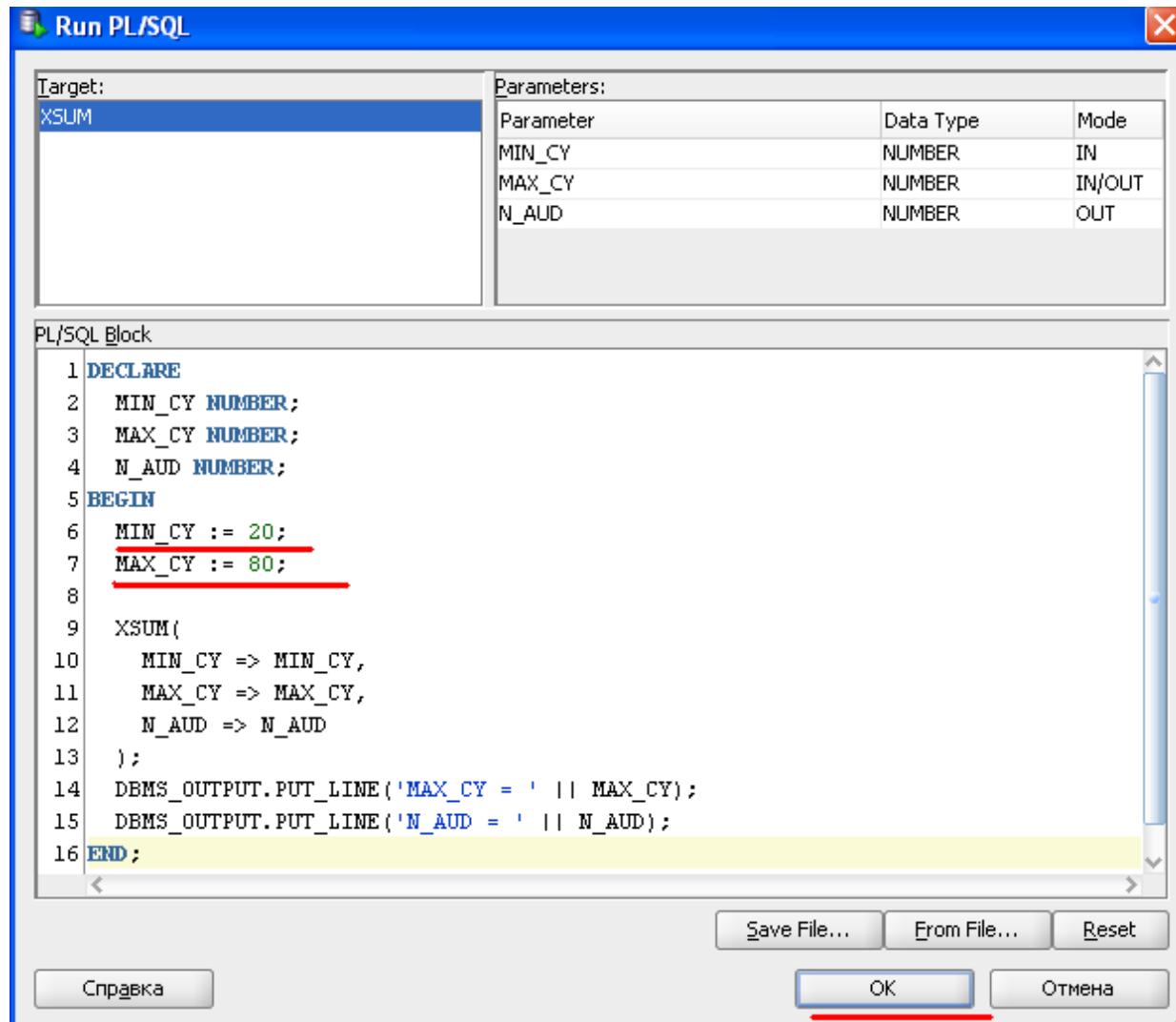
# Вызов процедуры



# Отладка



# Отладка



# Отладка

Running: IdeConnections%23shl-svvcore.jpr - Log X

Connecting to the database shl-svvcore.

MAX CY = 80

N AUD =

Process exited.

Disconnecting from the database shl-svvcore

# Переменные

```
-- 16/01.sql
--select * from dba_sys_privs where grantee = 'RLSVV';
-- select * from user_role_privs where username = 'SVVCORE';
-- grant create procedure to RLSVV

create or replace procedure svvcore.xsum(
    min_cy  in      svvcore.auditorium.auditorium_capacity%type, -- минимальная вместимость
    max_cy  in out   svvcore.auditorium.auditorium_capacity%type, -- максимальная вместимость
    n_aud   out      number                                -- количество
)
is
    m_max_cy  svvcore.auditorium.auditorium_capacity%type;
    m_n_aud   number := 0;
begin
    select count(*), max(auditorium_capacity) into m_n_aud, m_max_cy from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
    max_cy := m_max_cy;
    n_aud := m_n_aud;
exception
    when others then dbms_output.put_line(sqlerrm);
end xsum;
```



# Вызов процедур

```
:UUCORE@sh1> exec :max_cy:=10;
```

Процедура PL/SQL успешно завершена.

```
:UUCORE@sh1> exec xsum(20,:max_cy,:n_aud);
```

Процедура PL/SQL успешно завершена.

```
:UUCORE@sh1> select :max_cy, :n_aud from dual;
```

:MAX_CY	:N_AUD
-----	-----
0	
-----	-----

```
:UUCORE@sh1>
```

```
:UUCORE@sh1> var max_cy number;
```

```
:UUCORE@sh1> var n_aud number;
```

```
:UUCORE@sh1> var
```

переменная max\_cy

тип данных NUMBER

переменная n\_aud

тип данных NUMBER

```
:UUCORE@sh1> exec :max_cy:=20;
```

Процедура PL/SQL успешно завершена.

```
:UUCORE@sh1> exec :max_cy:=80;
```

Процедура PL/SQL успешно завершена.

```
:UUCORE@sh1> exec xsum(20,:max_cy,:n_aud);
```

Процедура PL/SQL успешно завершена.

```
:UUCORE@sh1> select :max_cy, :n_aud from dual;
```

:MAX_CY	:N_AUD
-----	-----
60	5
-----	-----



# Вызов процедур

```
-- 16/02.sql
--select * from dba_sys_privs where grantee = 'RLSVV';
-- select * from user_role_privs where username = 'SVVCORE';
-- grant create procedure to RLSVV

create or replace procedure svvcore.xsum(
    min_cy  in      svvcore.auditorium.auditorium_capacity%type, -- минимальная вместимость
    max_cy  in out   svvcore.auditorium.auditorium_capacity%type, -- максимальная вместимость
    n_aud   out      number                                -- количество
)
is
begin
    select count(*), max(auditorium capacity) into n_aud, max_cy
        from svvcore.auditorium
       where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
exception
    when others then dbms_output.put_line(sqlerrm);
end xsum;
```



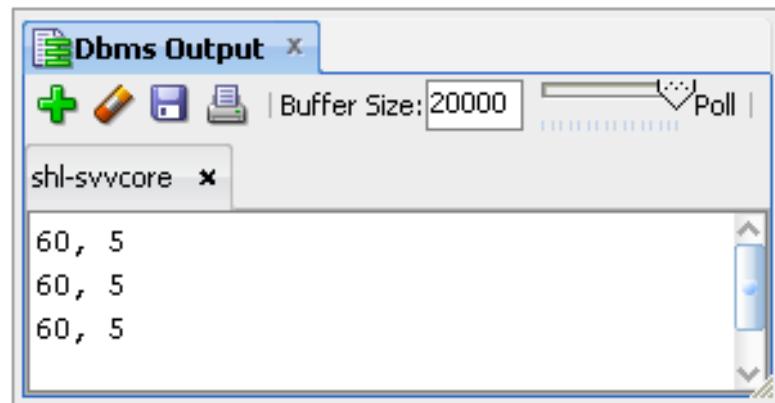
# Вызов процедур

```
-- 16/03.sql
declare
    max_cy number(3) := 80;
    n_aud  number(3):= 0;
begin
    svvcore.xsum(20, max_cy, n_aud);
    dbms_output.put_line(max_cy || ',' || n_aud);

    xsum(n_aud=>n_aud, min_cy =>20, max_cy=>max_cy);
    dbms_output.put_line(max_cy || ',' || n_aud);

    xsum(20, n_aud=>n_aud, max_cy=>max_cy);
    dbms_output.put_line(max_cy || ',' || n_aud);

end;
```



# Значения по умолчанию - DEFAULT

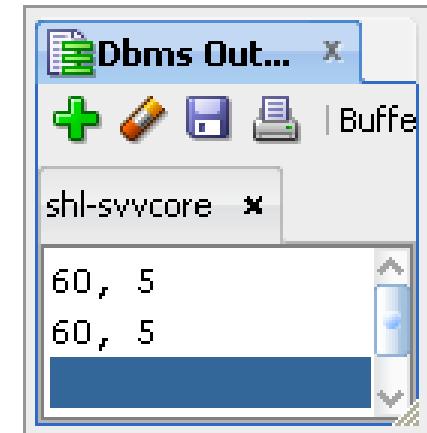
```
-- 16/04.sql
create or replace procedure svvcore.xsum(
    min_cy  in    svvcore.auditorium.auditorium_capacity%type default 20, -- минимальная вместимость
    max_cy  in out svvcore.auditorium.auditorium_capacity%type,      -- максимальная вместимость
    n_aud   out    number                                              -- количество
)
is
    no_max_cy exception;
begin
    select count(*), max(auditorium_capacity) into n_aud, max_cy from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
    if n_aud is null
    then raise no_max_cy;
    end if;
exception
    when no_max_cy then return;
    when others then dbms_output.put_line(sqlerrm);
end xsum;
```

# Значения по умолчанию - DEFAULT

```
-- 16/05.sql
declare
    max_cy number(3) := 80;
    n_aud  number(3):= 0;
begin
    --svvcore.xsum(max_cy, n_aud);
    --dbms_output.put_line(max_cy || ', ' || n_aud);
    xsum(n_aud=>n_aud, max_cy=>max_cy);
    dbms_output.put_line(max_cy || ', ' || n_aud);

    xsum(n_aud=>n_aud,  max_cy=>max_cy);
    dbms_output.put_line(max_cy || ', ' || n_aud);

end;
```



# Компиляция

---

- ▶ OR REPLACE – перестроение уже существующего модуля, привилегии на выполнение сохраняются
- ▶ AUTHID – определяет, как будет выполняться модуль и разрешаться имена в БД:
  - ▶ DEFINER – (по умолчанию) от имени владельца модуля
  - ▶ CURRENT\_USER - от имени пользователя, выполняющего модуль



# Вызов процедуры пользователем, не являющимся владельцем

```
SUVCORE@sh1> connect svvguest/svvguest@sh1;
Соединено.
sh1 - SVVGUEST - 12.12.10
SUVGUEST@sh1> var
переменная max_cy
типа данных NUMBER

переменная n_aud
типа данных NUMBER
SUVGUEST@sh1> exec :max_cy:=80;
Процедура PL/SQL успешно завершена.

SUVGUEST@sh1> exec xsum(20,:max_cy,:n_aud);
BEGIN xsum(20,:max_cy,:n_aud); END;

*
ошибка в строке 1:
ORA-06550: Стока 1, столбец 7:
PLS-00201: идентификатор 'XSUM' должен быть объявлен
ORA-06550: Стока 1, столбец 7:
PL/SQL: Statement ignored

SUVGUEST@sh1> exec suvcore.xsum(20,:max_cy,:n_aud);
BEGIN suvcore.xsum(20,:max_cy,:n_aud); END;

*
ошибка в строке 1:
ORA-06550: Стока 1, столбец 7:
PLS-00201: идентификатор 'SUVCORE.XSUM' должен быть объявлен
ORA-06550: Стока 1, столбец 7:
PL/SQL: Statement ignored
```



# AUTHID {CURRENT\_USER | DEFINER}

```
-- 16/07.sql
create or replace procedure svvcore.xxsum(
    min_cy  in      svvcore.auditorium.auditorium_capacity%type default 20, -- минимальная в
    max_cy  in out  svvcore.auditorium.auditorium_capacity%type,   -- максимальная вместимость
    n_aud   out     number                                         -- количество
)
authid current user is
    no_max_cy exception;
begin
    select count(*), max(auditorium_capacity) into n_aud, max_cy from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
    if n_aud is null
        then raise no_max_cy;
    end if;
exception
    when no_max_cy then return;
    when others then dbms_output.put_line(sqlerrm);
end xxsum;
```



# SQL-оператор CALL вызова процедур

```
SUUGUEST@sh1>
SUUGUEST@sh1>
SUUGUEST@sh1> CALL suvcore.xsum(20,:max_cy,:n_aud);
Вызов завершен.
SUUGUEST@sh1> select :max_cy, :n_aud from dual;

:MAX_CY      :N_AUD
----- -----
 60          5
```



# USER PROCEDURES

```
SUUGUEST@sh1> connect suvcore/suvcore@sh1;
```

Соединено.

sh1 - SUVCORE - 13.12.10

```
SUVCORE@sh1> select * from user_procedures;
```

OBJECT_NAME	PROCEDURE_NAME	AGG	PIP
XSUM		NO	NO
XXSUM		NO	NO
CALENDARPKG	XINSERT	NO	NO
CALENDARPKG	NSEMESTER	NO	NO
SUVCORE@sh1>			
SUVCORE@sh1>			

# USER\_SOURCE

```
SVVCORE@sh1> column text format a100
SVVCORE@sh1> column name format a10
SVVCORE@sh1> select type, name, line, text from user_source where name ='XSUM'
 2 /
```

TYPE	NAME	LINE	TEXT
PROCEDURE	XSUM	1	procedure xsum(
PROCEDURE	XSUM	2	min_cy in svvcore.auditorium.auditorium_capacity%type default 20, -- минимальная
			стимость
PROCEDURE	XSUM	3	max_cy in out svvcore.auditorium.auditorium_capacity%type, -- максимальная вместимость
PROCEDURE	XSUM	4	n_aud out number -- количество
PROCEDURE	XSUM	5	)
PROCEDURE	XSUM	6	is
PROCEDURE	XSUM	7	no_max_cy exception;
PROCEDURE	XSUM	8	begin
PROCEDURE	XSUM	9	select count(*), max(auditorium_capacity) into n_aud, max_cy from svvcore.auditorium
TYPE	NAME	LINE	TEXT
PROCEDURE	XSUM	10	where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy;
PROCEDURE	XSUM	11	if n_aud is null
PROCEDURE	XSUM	12	then raise no_max_cy;
PROCEDURE	XSUM	13	end if;
PROCEDURE	XSUM	14	exception
PROCEDURE	XSUM	15	when no_max_cy then return;
PROCEDURE	XSUM	16	when others then dbms_output.put_line(sqlerrm);
PROCEDURE	XSUM	17	end xsum;
PROCEDURE	XSUM	18	
PROCEDURE	XSUM	19	
PROCEDURE	XSUM	20	

20 строк выбрано.

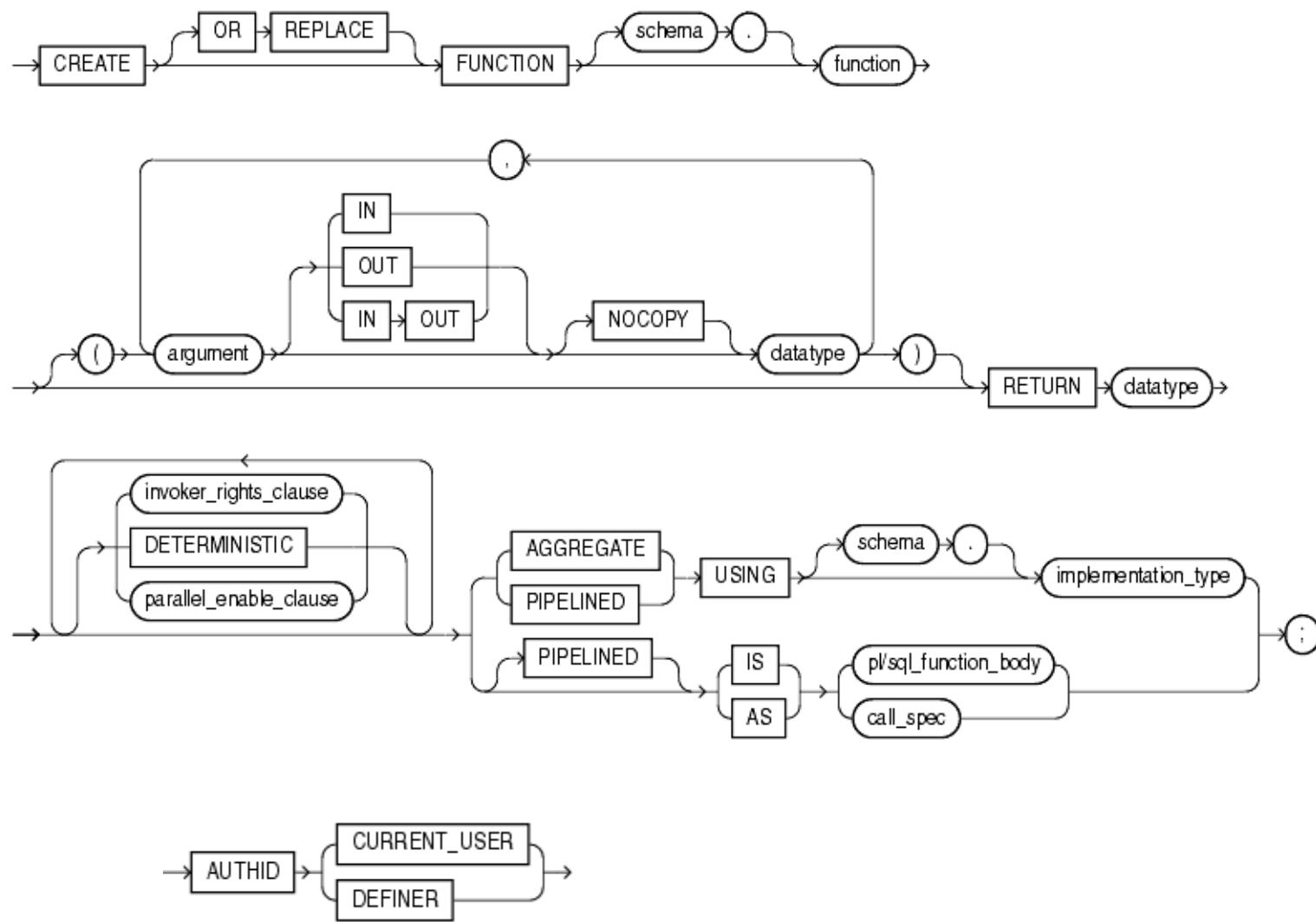
# Функция

---

- ▶ Функция – именованный модуль, который выполняет ноль или более выражений через фразу Return
- ▶ Может быть вызвана следующим образом:
  - ▶ В присвоении начального значения переменной
  - ▶ В выражении присвоения
  - ▶ В булевом выражении
  - ▶ В SQL запросе
  - ▶ Как аргумент в списке параметров другой функции или процедуры



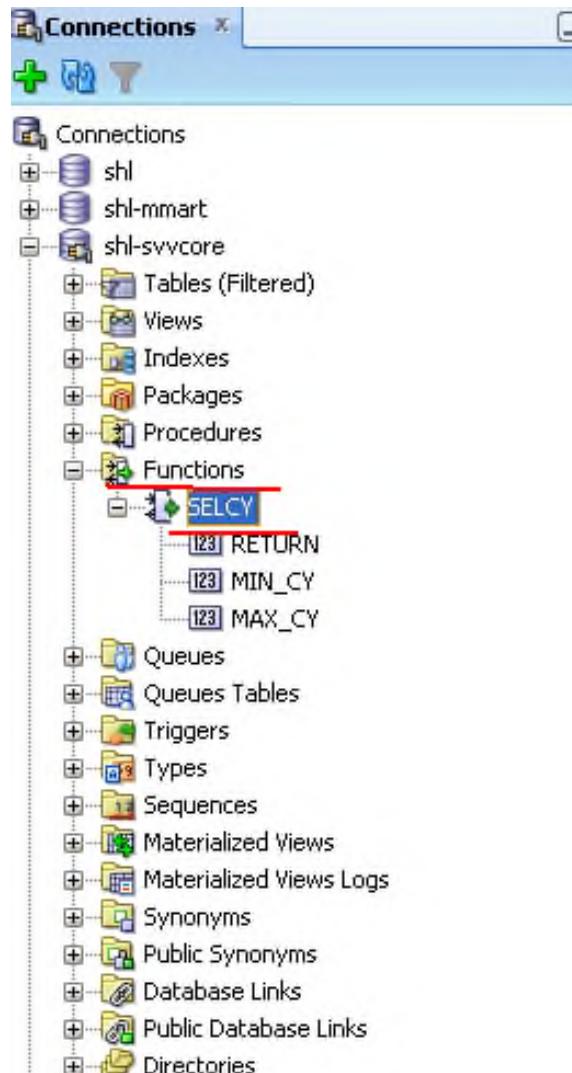
# ФУНКЦИИ



# ФУНКЦИЯ

```
-- 16/10 .sql
create or replace function selcy(
    min_cy auditorium.auditorium_capacity%type,
    max_cy in out auditorium.auditorium_capacity%type
)
return number is
    rc number(5);
begin
    select count(*), max(auditorium_capacity) into rc, max_cy from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <= max_cy group by max_cy;
    return rc;
exception
    when others then return -1;
end selcy;
```

# SQL Developer



# Отладка

Run PL/SQL

Target: SELCY

Parameter	Data Type	Mode
<Return Value>	NUMBER	OUT
MIN_CY	NUMBER	IN
MAX_CY	NUMBER	IN/OUT

PL/SQL Block

```
1 DECLARE
2   MIN_CY NUMBER;
3   MAX_CY NUMBER;
4   v_Return NUMBER;
5 BEGIN
6   MIN_CY := 20;
7   MAX_CY := 80;
8
9   v_Return := SELCY(
10    MIN_CY => MIN_CY,
11    MAX_CY => MAX_CY
12 );
13 DBMS_OUTPUT.PUT_LINE('MAX_CY = ' || MAX_CY);
14 DBMS_OUTPUT.PUT_LINE('v_Return = ' || v_Return);|
15 END;
16
```

Connecting to the database sh1-svvco  
MAX\_CY = 60  
v\_Return = 5  
Process exited.  
Disconnecting from the database sh1-svvco

Save File... From File... Reset OK Отмена Справка

# Применение функций в SELECT

```
SUVCORE@sh1>
SUVCORE@sh1> select selcy(20,80) from dual;
select selcy(20,80) from dual
*
ошибка в строке 1:
ORA-06572: Функция SELCY имеет внешний аргумент
```

```
SUVCORE@sh1>
SUVCORE@sh1>
SUVCORE@sh1>
SUVCORE@sh1> var max_cy number;
SUVCORE@sh1> exec :max_cy:=80;
```

Процедура PL/SQL успешно завершена.

```
SUVCORE@sh1> select selcy(20,:max_cy) from dual;
select selcy(20,:max_cy) from dual
*
ошибка в строке 1:
ORA-06572: Функция SELCY имеет внешний аргумент
```



# Функция без параметров

```
create or replace function maximumcy
return number is
    rc  number(5);
begin
    select max(auditorium_capacity) into rc from svvcore.auditorium;
    return rc;
exception
    when others then return -1;
end maximumcy;
```



# ВЫЗОВ В SELECT

```
-- 16/11 .sql
create or replace function maxcy(
    min_cy auditorium.auditorium_capacity%type,
    max_cy auditorium.auditorium_capacity%type
)
return number is
    rc  number(5);
begin
    select count(*) into rc from svvcore.auditorium
    where auditorium_capacity >= min_cy and auditorium_capacity <=  max_cy ;
    return rc;
exception
    when others then return -1;
end maxcy;
```

```
SVVCORE@sh1>
SVVCORE@sh1>
SVVCORE@sh1>
SVVCORE@sh1> select maxcy(20,80) from dual;
MAXCY(20,80)
-----
5
-----
```



# Ключевые слова

---

- ▶ **DETERMINISTIC** – функция детерминирована, если она возвращает одно и то же значение при вызове с теми же параметрами
- ▶ **AGGREGATE USING** – используется для агрегатных функций.



# DETERMINISTIC

---

```
-- 16/13 .sql
create or replace function maximumy_d
  return number deterministic is
    rc number(5);
begin
  select max(auditorium_capacity) into rc from svvcore.auditorium;
  return rc;
exception
  when others then return -1;
end maximumy_d;
```



# Пакеты

---

- ▶ Пакеты - коллекция PL/SQL объектов, сгруппированных вместе.
- ▶ Преимущества:
  - ▶ Скрытие информации
  - ▶ Объектно-ориентированный дизайн
  - ▶ Постоянство объектов в транзакциях
  - ▶ Улучшенная производительность
- ▶ Можно включать в пакет: процедуры, функции, константы, исключения, курсоры, переменные, TYPE выражения, записи, REF курсоры



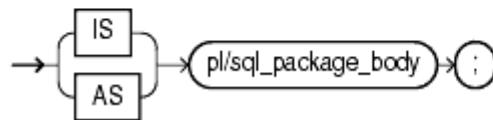
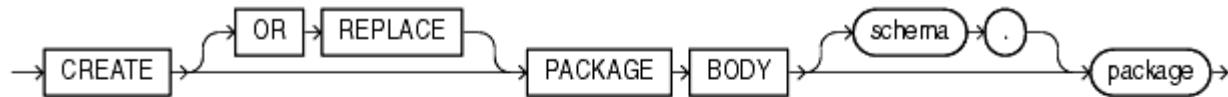
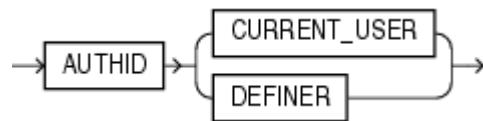
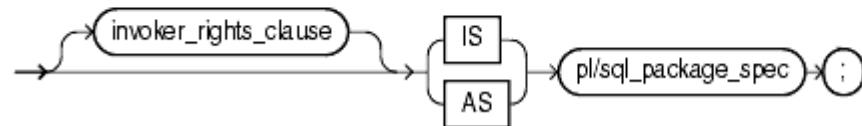
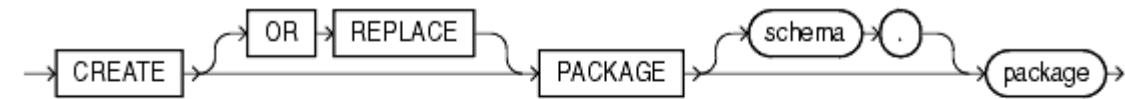
# Пакеты

---

- ▶ Спецификация пакета (*package*) – обязательна, содержит список объектов для общего доступа из других модулей или приложения
- ▶ Реализация пакета (*package body*) – содержит весь программный код для реализации процедур и функций и спецификации, приватные объекты и секцию инициализации



# Спецификация пакета



# Пример заголовка пакета

```
create or replace
package teacherpkg as

type teacher_rec is record
(
    t          teacher.teacher%type,
    tn         teacher.teacher_name%type,
    pp         teacher.pulpit%type
);

exc_xinsert exception;
exc_xupdate exception;
exc_xdelete exception;
procedure xinsert(tr teacher_rec);
function xupdate(tr teacher_rec) return boolean;
function xdelete(t teacher.teacher%type) return boolean;

end teacherpkg;
--- ===== ---
```



# Пример использования пакета

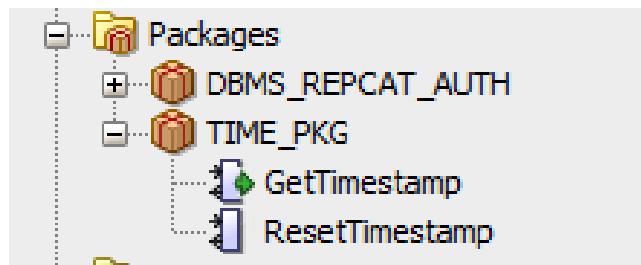
```
declare
    rec teacherpkg.teacher_rec;
begin
    rec.t := 'БНР';
    rec.tn := 'Бендер Остал Ибраимович';
    rec.pp := 'ИСиТ';
    teacherpkg.xinsert(rec);
    commit;
    rec.tn := 'Бурнаш Михаил Антонович';
    if teacherpkg.xupdate(rec)
        then dbms_output.put_line('xupdate = ok');
        else dbms_output.put_line('xupdate = error');
    end if;
    commit;

    if teacherpkg.xdelete(rec.t)
        then dbms_output.put_line('xdelete = ok');
        else dbms_output.put_line('xdelete = error');
    end if;
    commit;
exception
    when teacherpkg.exc_xdelete then dbms_output.put_line('xdelete:'||sqlerrm);
    when teacherpkg.exc_xinsert then dbms_output.put_line('xinsert:'||sqlerrm);
    when teacherpkg.exc_xupdate then dbms_output.put_line('xupdate:'||sqlerrm);

end;
```

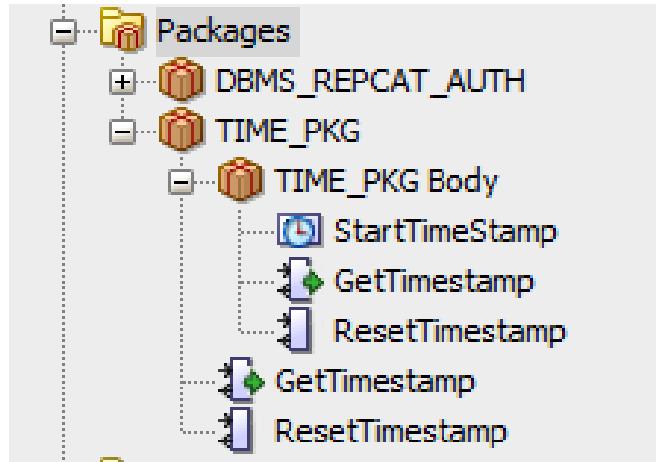
# Пример спецификации пакета

```
-- PACKAGES
CREATE OR REPLACE PACKAGE time_pkg IS
    FUNCTION GetTimestamp RETURN DATE;
    PROCEDURE ResetTimestamp(new_time DATE DEFAULT SYSDATE);
END time_pkg;
```



# Пример реализации пакета

```
CREATE OR REPLACE PACKAGE BODY time_pkg IS
    StartTimeStamp DATE := SYSDATE;
    -- StartTimeStamp is package data.
-- Function
FUNCTION GetTimestamp RETURN DATE IS
BEGIN
    RETURN StartTimeStamp;
END GetTimestamp;
-- Procedures
PROCEDURE ResetTimestamp(new_time DATE DEFAULT SYSDATE)
IS
BEGIN
    StartTimeStamp := new_time;
END ResetTimestamp;
-- Initialization section
BEGIN
    null;
EXCEPTION
    WHEN NO_DATA_FOUND
        THEN dbms_output.put_line('not initialized');
END time_pkg;
```



# Пример использования пакета

```
-- package use
begin
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));

    dbms_lock.sleep(3);

    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
    time_pkg.resetTimestamp;
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
end;
```

14/12/2019 08:55:59  
14/12/2019 08:55:59  
14/12/2019 08:56:02

```
-- package use
begin
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
    dbms_lock.sleep(3);
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
end;
```

14/12/2019 08:56:02  
14/12/2019 08:56:02  
14/12/2019 08:56:02  
14/12/2019 08:56:02

# Пример пакета SERIALLY REUSABLE

```
-- PACKAGES
CREATE OR REPLACE PACKAGE time_pkg IS
    PRAGMA SERIALLY_REUSABLE;
    FUNCTION GetTimestamp RETURN DATE;
    PROCEDURE ResetTimestamp(new_time DATE DEFAULT SYSDATE);
END time_pkg;

CREATE OR REPLACE PACKAGE BODY time_pkg IS
    PRAGMA SERIALLY_REUSABLE;
    StartTimeStamp DATE := SYSTIMESTAMP;
    -- StartTimeStamp is package data.
    -- Function
    FUNCTION GetTimestamp RETURN DATE IS
        BEGIN
            RETURN StartTimeStamp;
        END GetTimestamp;
    -- Procedures
    PROCEDURE ResetTimestamp(new_time DATE DEFAULT SYSDATE)
        IS
        BEGIN
            StartTimeStamp := new_time;
        END ResetTimestamp;
    -- Initialization section
    BEGIN
        null;
    EXCEPTION
        WHEN NO_DATA_FOUND
            THEN dbms_output.put_line('not initialized');
    END time_pkg;
```

# Пример пакета SERIALLY REUSABLE

```
-- package use
begin
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
    dbms_lock.sleep(3);
    dbms_output.put_line(to_char(time_pkg.gettimestamp, 'dd/mm/yyyy hh24:mi:ss'));
end;
```

14/12/2019 08:58:55

14/12/2019 08:58:55

14/12/2019 08:59:08

14/12/2019 08:59:08

# Пакеты

---

- ▶ Вызов пакета:
  - ▶ `Package_name.package_element;`
- ▶ Структуры данных, объявленные в пакете, называются пакетными данными
- ▶ Пакетные переменные сохраняют свое состояние от одной транзакции к другой и являются глобальными данными



# Пакеты

---

- ▶ AUTHID {CURRENT\_USER|DEFINER}
- ▶ Словарь: USER\_PROCEDURES, USER\_SOURCE
- ▶ ALTER PACKAGE COMPILE PACKAGE
- ▶ ALTER PACKAGE COMPILE BODY
- ▶ DROP PACKAGE



# Вопросы?

---



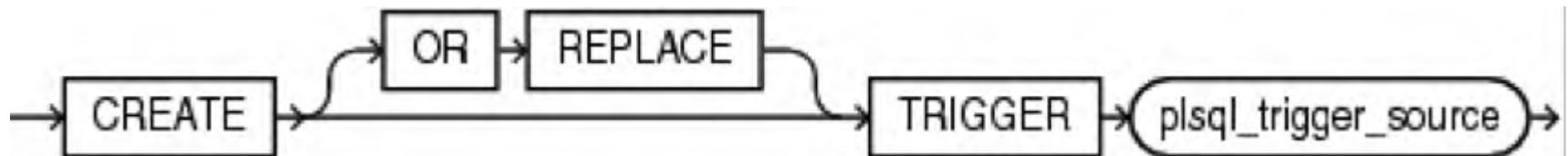
# **ORACLE 12c**

**PL/SQL Триггеры**

**Лекция 13**

# Триггеры

- ▶ Триггер – особый вид процедур, которые срабатывают по запускающему их событию



# Применение триггеров

---

- ▶ для реализации сложных ограничений целостности базы данных;
- ▶ для аудита (контроля хранимой и изменяемой информации);
- ▶ для автоматического оповещения программ о произошедших событиях;



# Привилегии

---

- ▶ **CREATE TRIGGER** - создавать, удалять, изменять в своей подсхеме
- ▶ **CREATE ANY TRIGGER** - создать любой триггер в любой схеме, кроме SYS, не рекомендуется для словаря, не разрешает менять текст триггера
- ▶ **ALTER ANY TRIGGER** - разрешать, запрещать, изменять, компилировать, любые, кроме SYS-триггеров, триггеры
- ▶ **DROP ANY TRIGGER** - удалять любой триггер, кроме SYS-триггеров
- ▶ **ADMINISTER DATABASE TRIGGER** - создавать, изменять, удалять системные триггеры, должен иметь привилегию CREATE TRIGGER или CREATE ANY TRIGGER



# Привилегии

---

```
SYSTEM@sh1>
SYSTEM@sh1> connect system/system@sh1;
Соединено.
sh1 - SYSTEM - 24.02.11
SYSTEM@sh1> grant create trigger to svvcore;
```

Привилегии предоставлены.



# Привилегии

---

- ▶ Триггеры выполняется под правами создателя триггера
- ▶ Назначаются напрямую USERу, а не через роль



# Транзакции

---

- ▶ Триггер – часть транзакции, ошибка в триггере откатывает операцию, изменения таблиц в триггере становятся частью транзакции.
- ▶ Если откатывается транзакция, изменения триггера тоже откатываются.
- ▶ Не может выдавать COMMIT/ROLBACK (исключение - только, если в теле триггера есть автономная транзакция)
- ▶ Может выдавать RAISE\_APPLICATION\_ERROR



# Автономные транзакции

```
create table dbst_t ( msg varchar2(25) );

create or replace procedure dbst_NonAutonomous_Insert
as
begin
    insert into dbst_t values ( 'NonAutonomous Insert' );
    commit;
end;

create or replace procedure dbst_Autonomous_Insert
as
pragma autonomous_transaction;
begin
    insert into dbst_t values ( 'Autonomous Insert' );
    commit;
end;
```

```
begin
    insert into dbst_t values ( 'Anonymous Block' );
    dbst_NonAutonomous_Insert;
    rollback;
end;
```

Anonymous Block  
NonAutonomous Insert

```
begin
    insert into dbst_t values ( 'Anonymous Block' );
    dbst_Autonomous_Insert;
    rollback;
end;
```

Autonomous Insert



# Триггеры

---

- ▶ DML-триггеры
- ▶ Системные триггеры



# Классификация триггеров

---

- ▶ **По привязанному объекту:**  
На таблице
- ▶ На представлении - instead of trigger
- ▶ **По событиям запуска:**  
Вставка записей - insert
- ▶ Обновление записей - update
- ▶ Удаление записей - delete
- ▶ **По области действия:**  
Уровень оператора - statement level triggers
- ▶ Уровень записи - row level triggers
- ▶ Составные триггеры - compound triggers
- ▶ **По времени срабатывания:**  
Перед выполнением операции – before
- ▶ После выполнения операции - after



# DML-триггеры

---

- ▶ Время события:
  - ▶ AFTER (после события) – после записи в журнал,
  - ▶ BEFORE (до события) – до записи в журнал;



# Уровни триггеров

---

- ▶ FOR EACH ROW (для каждой строки) - срабатывает для каждой измененной строки
- ▶ ПО УМОЛЧАНИЮ (операторный уровень) - срабатывает один раз на триггерное событие



# Порядок выполнения DML-триггеров

---

- ▶ операторные BEFORE;
- ▶ для каждой строки BEFORE;
- ▶ выполняется оператор;
- ▶ для каждой строки AFTER;
- ▶ операторные AFTER.



# Триггерные события DML

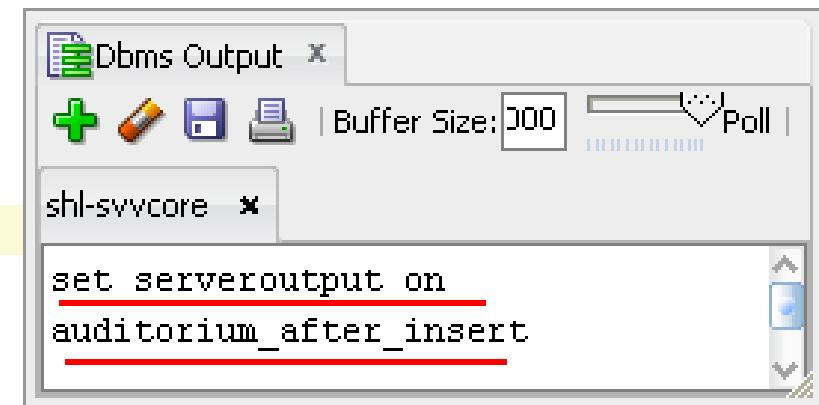
INSERT	Событие возникает, когда добавляется строка (строки) в таблицу или представление.
UPDATE	Событие возникает, когда выполняется операция UPDATE над данными в таблице или представлении. Можно дополнительно задавать выражение <b>OF</b> для указания полей, при изменении которых срабатывает триггер.
DELETE	Событие возникает, когда удаляется строка (строки) из таблицы или представления. Триггер <u>не срабатывает</u> при выполнении команды TRUNCATE table.



# Создание триггера на вставку

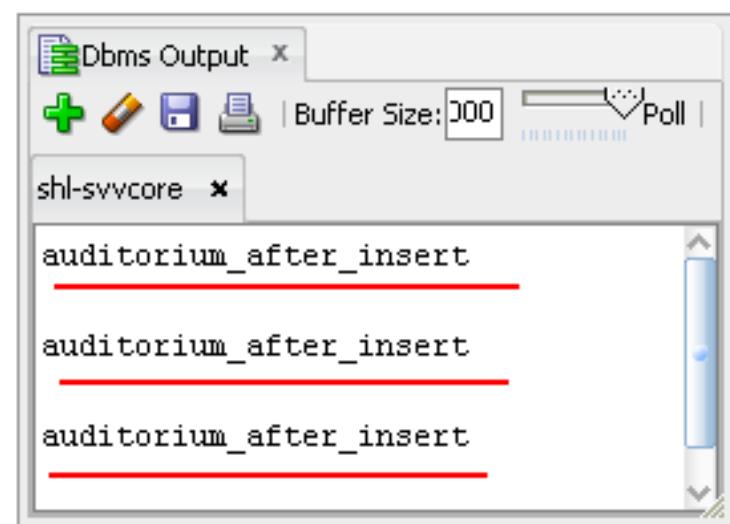
```
create or replace trigger auditorium_after_insert
after insert on auditorium
begin
    dbms_output.put_line('auditorium_after_insert');
end;

insert into auditorium(auditorium, auditorium_type, auditorium_capacity)
values ('324-1', 'ЛК', 60);
```



# Выполнение триггера

```
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('313-1', 'ЛК', 80);
insert into auditorium(auditorium, auditorium type,auditorium capacity)
    values ('222-4', 'ЛК', 60);
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('114-4', 'ЛК', 90);
```



# Триггер на обновление

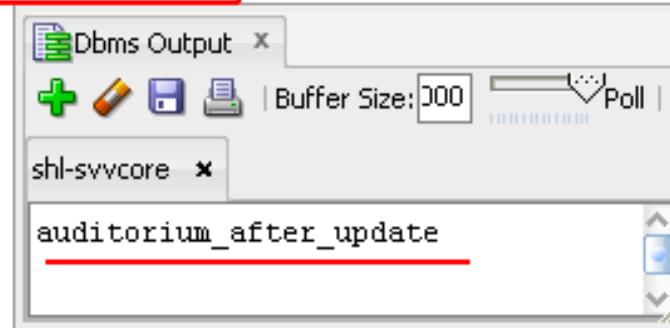
```
create or replace trigger auditorium after update  
after update on auditorium  
begin  
    dbms_output.put_line('auditorium_after_update');  
end;
```

```
select count(*) from auditorium;
```

COUNT(*)
4

```
update auditorium set auditorium_capacity = 100;
```

4 rows updated

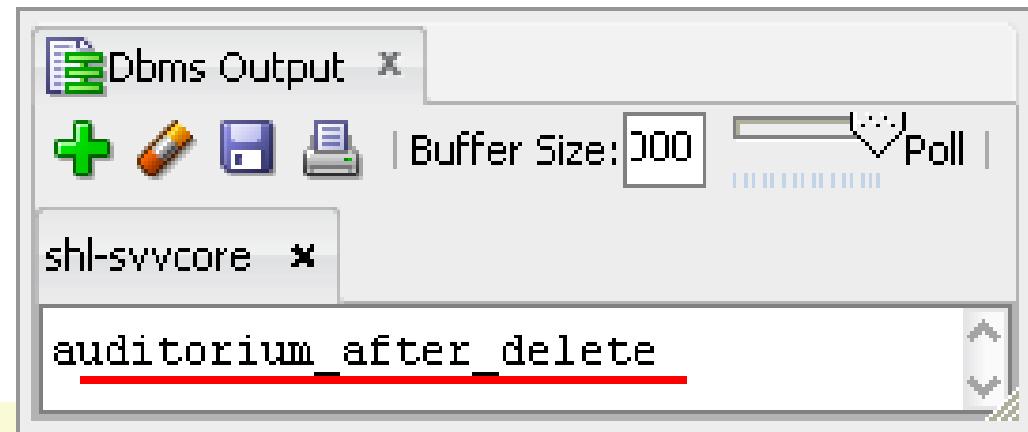


# Триггер на удаление

```
create or replace trigger auditorium_after_delete  
after delete on auditorium  
begin  
    dbms_output.put_line('auditorium_after_delete');  
end;
```

```
delete auditorium;
```

```
4 rows deleted
```

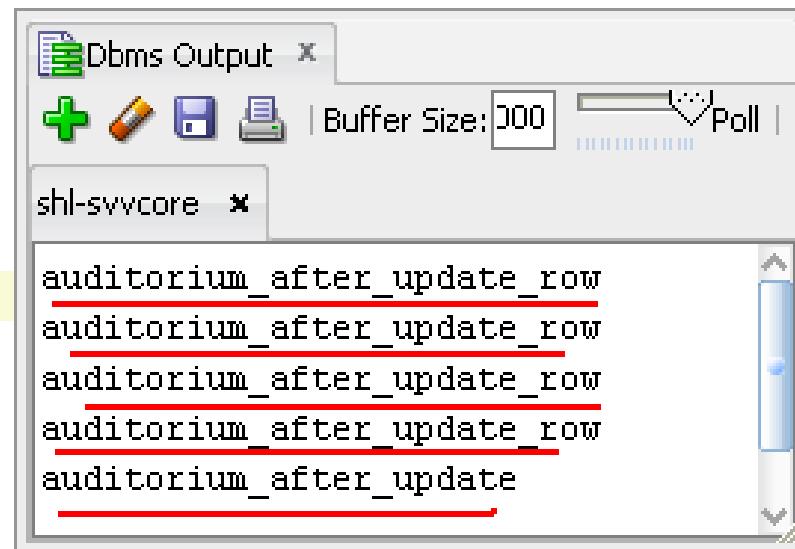


# Триггеры for each row

```
create or replace trigger auditorium_after_update_row
after update on auditorium
for each row
begin
    dbms_output.put_line('auditorium_after_update_row');
end;
```

```
update auditorium set auditorium_capacity = 120;
```

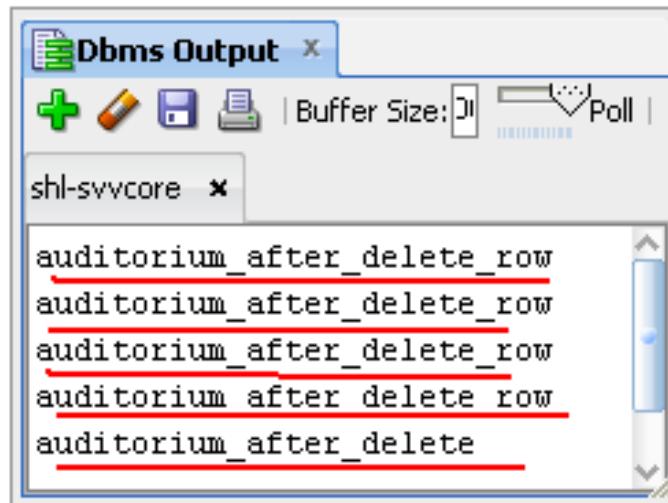
```
4 rows updated
```



# Триггеры for each row

```
create or replace trigger auditorium_after_delete_row
after delete on auditorium
for each row
begin
    dbms_output.put_line('auditorium_after_delete_row');
end;

delete auditorium;
4 rows deleted
```



# Предикаты триггера

---

- ▶ Чтобы различать DML команды и события, которые выполняют триггер, используются триггерные предикаты INSERTING, UPDATING, and DELETING в условиях IF

```
drop trigger auditorium_after_insert;
drop trigger auditorium_after_update;
drop trigger auditorium_after_delete;

create or replace trigger auditorium_after_1
after insert or update or delete  on auditorium
begin
  if inserting then
    dbms_output.put_line('auditorium_after_insert_1');
  elsif updating then
    dbms_output.put_line('auditorium_after_update_1');
  elsif deleting then
    dbms_output.put_line('auditorium_after_delete_1');
  end if;
end;
```



# Предикаты триггера

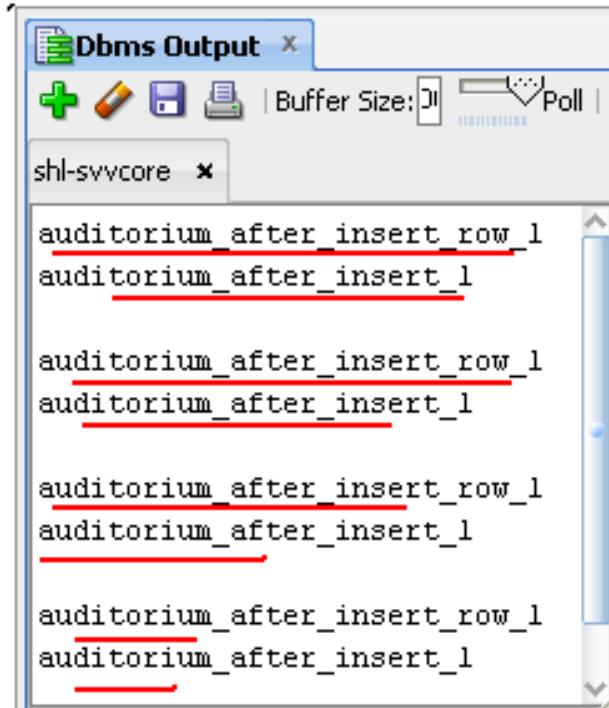
## ▶ Для триггера for each row

```
drop trigger auditorium_after_update_row;
drop trigger auditorium_after_delete_row;

create or replace trigger auditorium_after_row_1
after insert or update or delete  on auditorium
for each row
begin
  if inserting then
    dbms_output.put_line('auditorium_after_insert_row_1');
  elsif updating then
    dbms_output.put_line('auditorium_after_update_row_1');
  elsif deleting then
    dbms_output.put_line('auditorium_after_delete_row_1');
  end if;
end;
```

# Применение набора триггеров

```
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('324-1', 'ЛК', 60);
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('313-1', 'ЛК', 80);
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('222-4', 'ЛК', 60);
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
    values ('114-4', 'ЛК', 90);
```



The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it are icons for creating a new connection, editing, saving, and printing. There is also a "Buffer Size:" dropdown and a "Poll" button. The main pane displays four identical log entries, each consisting of two lines: "auditorium\_after\_insert\_row\_1" and "auditorium\_after\_insert\_1". The entire window is framed by a blue border.

```
Dbms Output x
+ 🖊️ 📁 🖨️ | Buffer Size: 0 | Poll |
shl-svvcore x
auditorium_after_insert_row_1
auditorium_after_insert_1

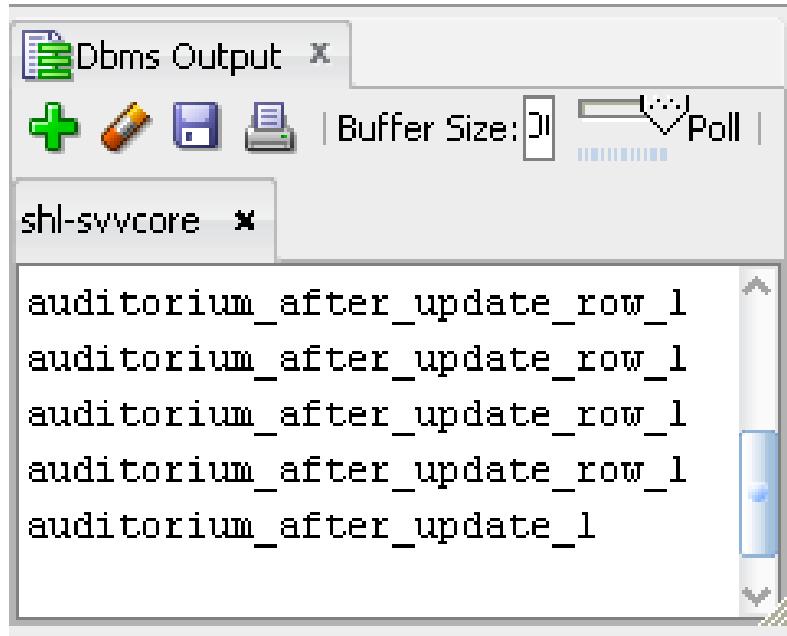
auditorium_after_insert_row_1
auditorium_after_insert_1

auditorium_after_insert_row_1
auditorium_after_insert_1

auditorium_after_insert_row_1
auditorium_after_insert_1
```

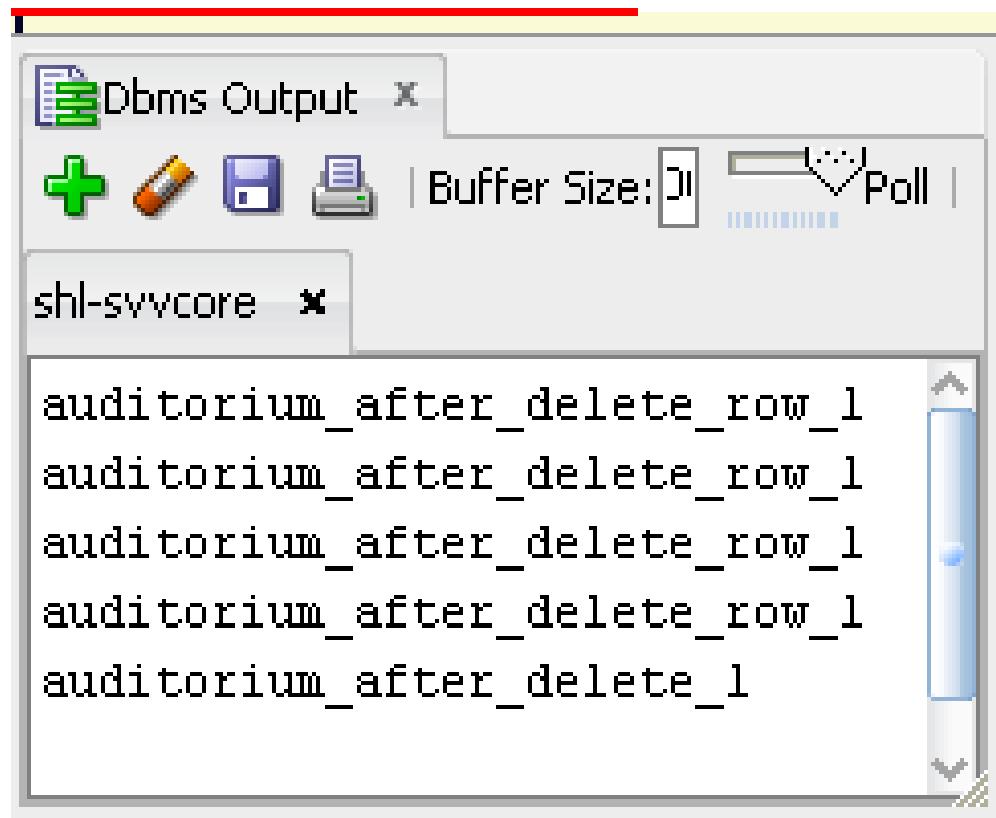
# Применение набора триггеров

update auditorium set auditorium\_capacity = 120;



# Применение набора триггеров

```
delete auditorium;
```



# Порядок выполнения триггеров

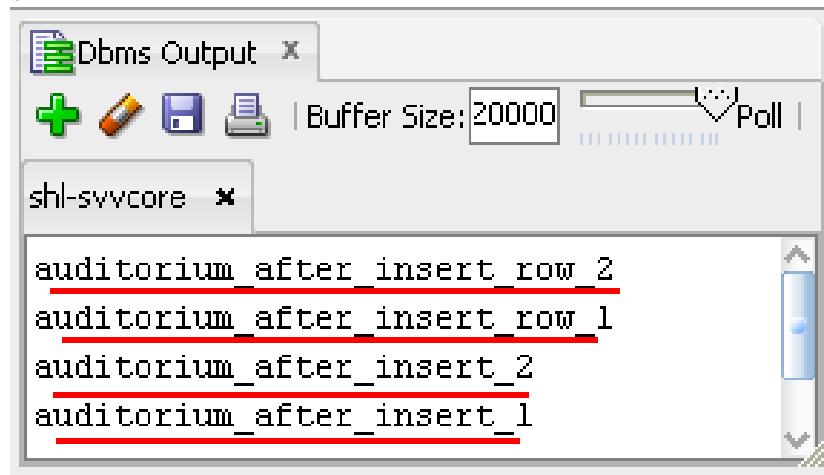
```
create or replace trigger auditorium_after_2
after insert or update or delete  on auditorium
begin
    if inserting then
        dbms_output.put_line('auditorium_after_insert_2');
    elsif updating then
        dbms_output.put_line('auditorium_after_update_2');
    elsif deleting then
        dbms_output.put_line('auditorium_after_delete_2');
    end if;
end;

create or replace trigger auditorium_after_row_2
after insert or update or delete  on auditorium
for each row
begin
    if inserting then
        dbms_output.put_line('auditorium_after_insert_row_2');
    elsif updating then
        dbms_output.put_line('auditorium_after_update_row_2');| 
    elsif deleting then
        dbms_output.put_line('auditorium_after_delete_row_2');
    end if;
end;
```



# Порядок выполнения триггеров

```
insert into auditorium(auditorium, auditorium_type,auditorium_capacity)
values ('301-4', 'ПК', 100);
```

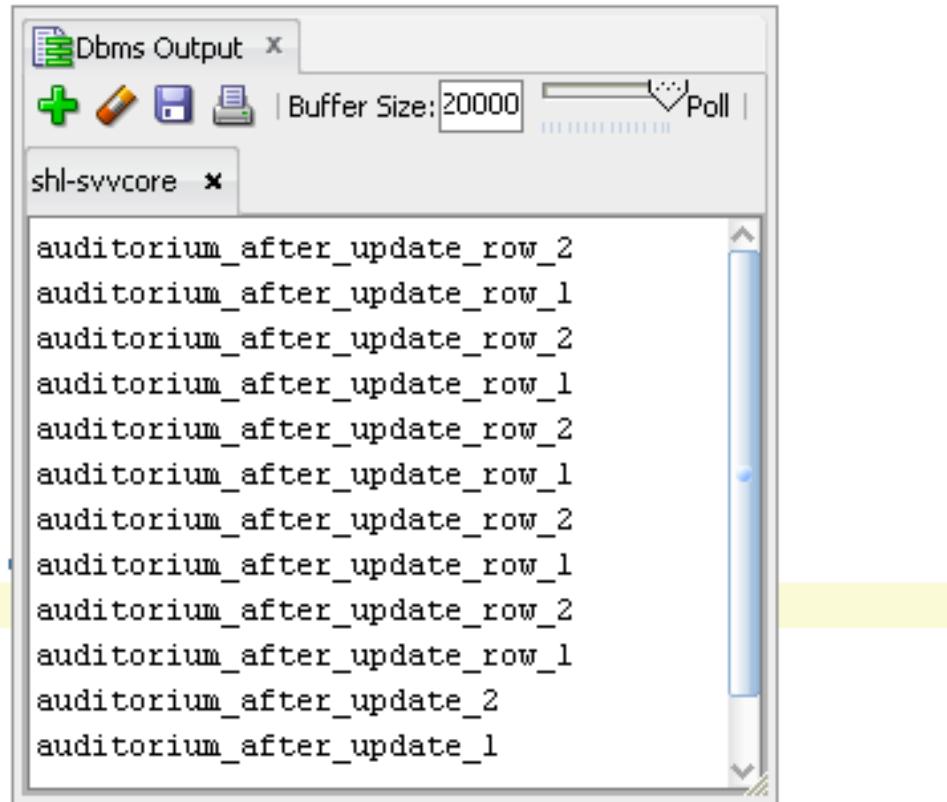


The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it, there are icons for a plus sign, a pencil, a floppy disk, and a printer. A dropdown menu is open, showing "Buffer Size: 20000" and a "Poll" checkbox. The main pane displays the following text:

```
shl-svvcore
auditorium_after_insert_row_2
auditorium_after_insert_row_1
auditorium_after_insert_2
auditorium_after_insert_1
```

# Порядок выполнения триггеров

```
update auditorium set auditorium_capacity = 120;
```



# Before - триггеры

---

```
create or replace trigger auditorium_before_1
before insert or update or delete on auditorium
begin
  if inserting then
    dbms_output.put_line('auditorium_before_insert_1');
  elsif updating then
    dbms_output.put_line('auditorium_before_update_1');
  elsif deleting then
    dbms_output.put_line('auditorium_before_delete_1');
  end if;
end;
```

```
create or replace trigger auditorium_before_row_1
before insert or update or delete on auditorium
for each row
begin
  if inserting then
    dbms_output.put_line('auditorium_before_insert_row_1');
  elsif updating then
    dbms_output.put_line('auditorium_before_update_row_1');
  elsif deleting then
    dbms_output.put_line('auditorium_before_delete_row_1');
  end if;
end;
```

---



# Before - триггеры

---

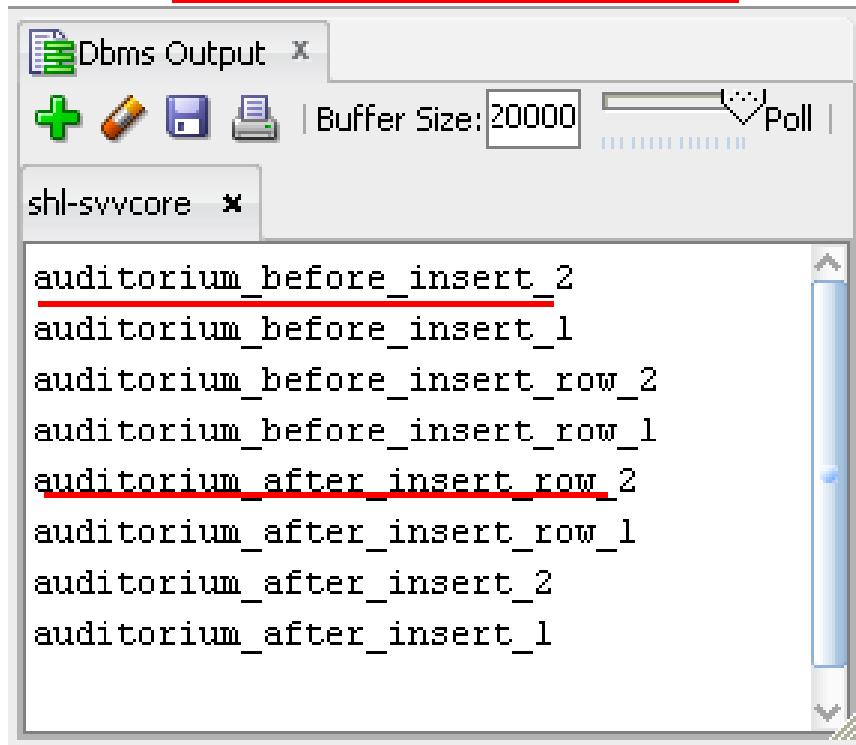
```
create or replace trigger auditorium_before_2
before insert or update or delete on auditorium
begin
    if inserting then
        dbms_output.put_line('auditorium_before_insert_2');
    elsif updating then
        dbms_output.put_line('auditorium_before_update_2');
    elsif deleting then
        dbms_output.put_line('auditorium_before_delete_2');
    end if;
end;

create or replace trigger auditorium_before_row_2
before insert or update or delete on auditorium
for each row
begin
    if inserting then
        dbms_output.put_line('auditorium_before_insert_row_2');
    elsif updating then
        dbms_output.put_line('auditorium_before_update_row_2');
    elsif deleting then
        dbms_output.put_line('auditorium_before_delete_row_2');
    end if;
end;
```



# Before - триггеры

```
insert into auditorium(auditorium, auditorium type,auditorium capacity)
values ('137-4', 'ЛК', 60);
```



The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it, there are icons for creating a new connection (+), editing, saving, and printing. A "Buffer Size" dropdown is set to 20000, and a "Poll" button is visible. The main pane displays a list of trigger names:

- auditorium\_before\_insert\_2
- auditorium\_before\_insert\_1
- auditorium\_before\_insert\_row\_2
- auditorium\_before\_insert\_row\_1
- auditorium\_after\_insert\_row\_2
- auditorium\_after\_insert\_row\_1
- auditorium\_after\_insert\_2
- auditorium\_after\_insert\_1

# Before - триггеры

```
update auditorium set auditorium_capacity = 100;
```

```
Dbms Output x  
+ 🖊 Buffer Size: 20000 Poll |  
shl-svvcore x  
auditorium_before_update_2  
auditorium_before_update_1  
auditorium_before_update_row_2  
auditorium_before_update_row_1  
auditorium_after_update_row_2  
auditorium_after_update_row_1  
auditorium_after_update_2  
auditorium_after_update_1
```

# Псевдозаписи new, old

Операция срабатывания триггера	OLD.column	NEW.column
Insert	Null	Новое значение
Update	Старое значение	Новое значение
Delete	Старое значение	Null



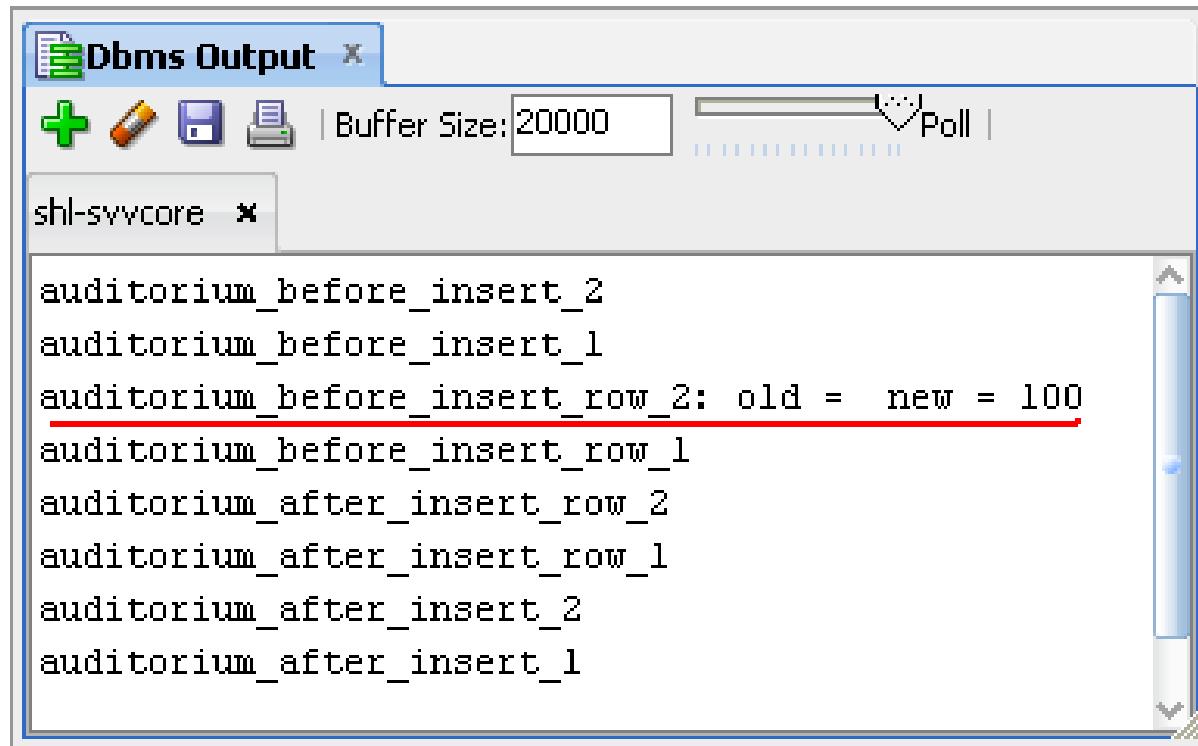
# Псевдозаписи new, old

```
create or replace trigger auditorium_before_row_2
before insert or update or delete  on auditorium
for each row
begin
  if inserting then
    dbms_output.put_line
      ('auditorium_before_insert_row_2:'
       ||' old = '|| :old.auditorium_capacity
       ||' new = '|| :new.auditorium_capacity
      );
  elsif updating then
    dbms_output.put_line
      ('auditorium_before_update_row_2:'
       ||' old = '|| :old.auditorium_capacity
       ||' new = '|| :new.auditorium_capacity
      );
  elsif deleting then
    dbms_output.put_line
      ('auditorium_before_delete_row_2:'
       ||' old = '|| :old.auditorium_capacity
       ||' new = '|| :new.auditorium_capacity
      );
  end if;
end;
```



# Псевдозаписи new, old

```
insert into auditorium(auditorium, auditorium type, auditorium capacity)
values ('405-1', 'ЛК', 100);
```

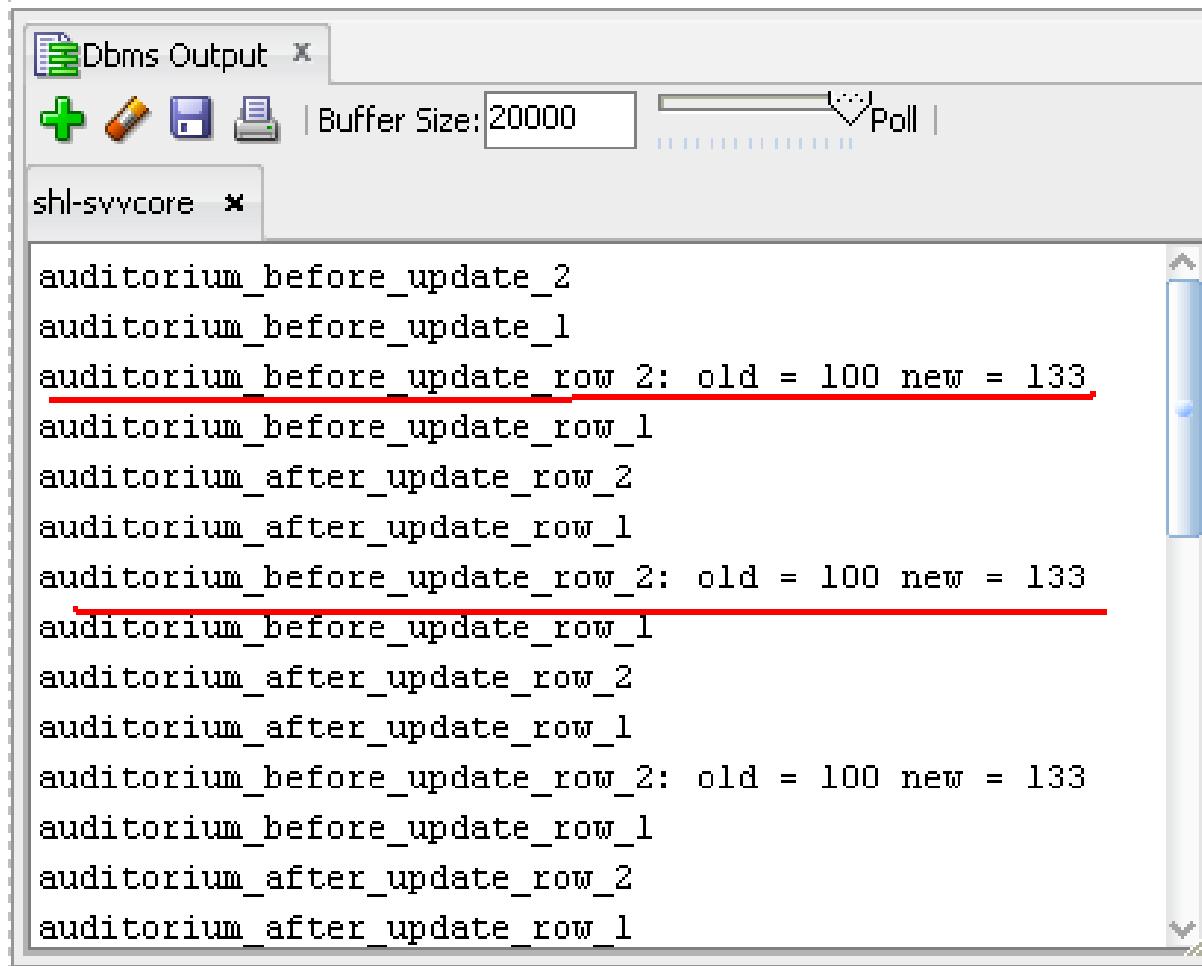


The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it is a toolbar with icons for New (+), Edit (pencil), Save (disk), Print (printer), Buffer Size (set to 20000), and Poll. The main area displays log messages for an insert operation into the auditorium table:

```
shl-svycore x
auditorium_before_insert_2
auditorium_before_insert_1
auditorium_before_insert_row_2: old =  new = 100
auditorium_before_insert_row_1
auditorium_after_insert_row_2
auditorium_after_insert_row_1
auditorium_after_insert_2
auditorium_after_insert_1
```

# Псевдозаписи new, old

```
update auditorium set auditorium_capacity = 133;
```



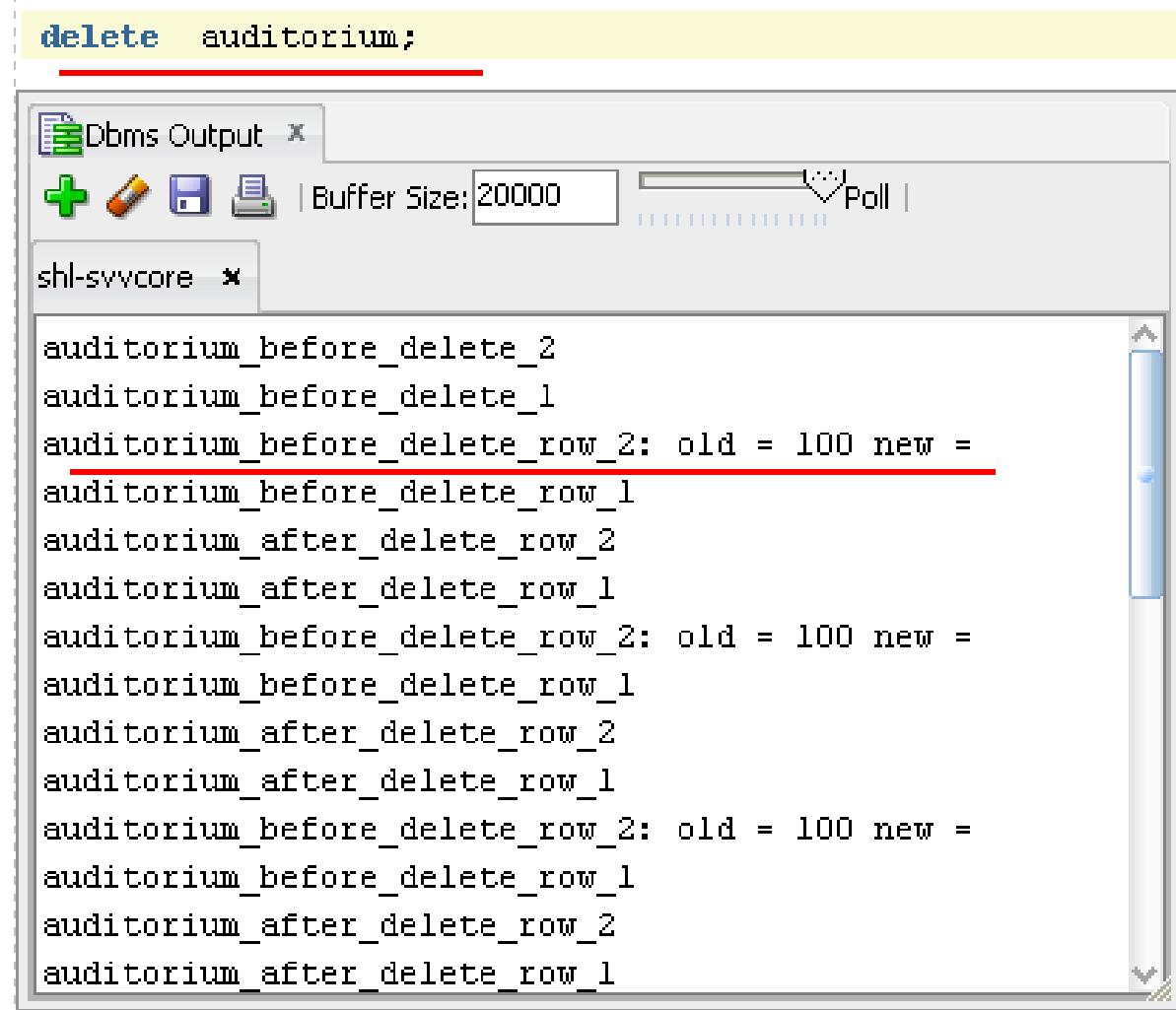
The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it are icons for New (+), Edit (pencil), Save (disk), and Print (printer). A "Buffer Size: 20000" input field and a "Poll" button are also present. The main area displays the following log messages:

```
shl-svycore *  
auditorium_before_update_2  
auditorium_before_update_1  
auditorium_before_update_row_2: old = 100 new = 133  
auditorium_before_update_row_1  
auditorium_after_update_row_2  
auditorium_after_update_row_1  
auditorium_before_update_row_2: old = 100 new = 133  
auditorium_before_update_row_1  
auditorium_after_update_row_2  
auditorium_after_update_row_1  
auditorium_before_update_row_2: old = 100 new = 133  
auditorium_before_update_row_1  
auditorium_after_update_row_2  
auditorium_after_update_row_1
```

The message "auditorium\_before\_update\_row\_2: old = 100 new = 133" is underlined twice, once by a red line and once by a blue line, highlighting the pseudorecord being updated.

# Псевдозаписи new, old

```
delete auditorium;
```

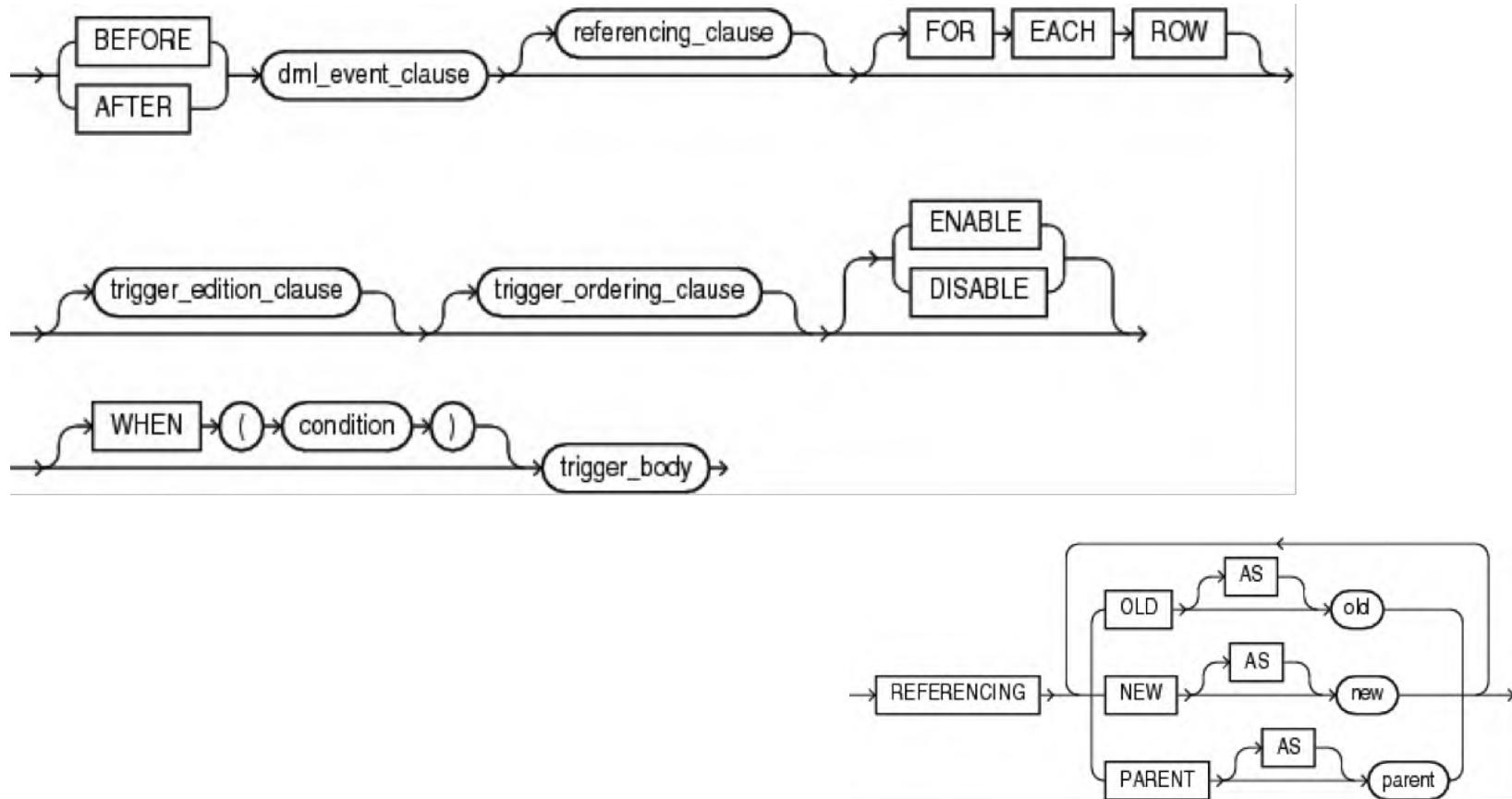


The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it are icons for New (+), Edit (pencil), Save (disk), and Print (printer). A "Buffer Size" dropdown is set to 20000. There is also a "Poll" button with a progress bar. The main area displays a log of database events:

```
shl-svvcore *  
  
auditorium_before_delete_2  
auditorium_before_delete_1  
auditorium_before_delete_row_2: old = 100 new =  
auditorium_before_delete_row_1  
auditorium_after_delete_row_2  
auditorium_after_delete_row_1  
auditorium_before_delete_row_2: old = 100 new =  
auditorium_before_delete_row_1  
auditorium_after_delete_row_2  
auditorium_after_delete_row_1  
auditorium_before_delete_row_2: old = 100 new =  
auditorium_before_delete_row_1  
auditorium_after_delete_row_2  
auditorium_after_delete_row_1
```

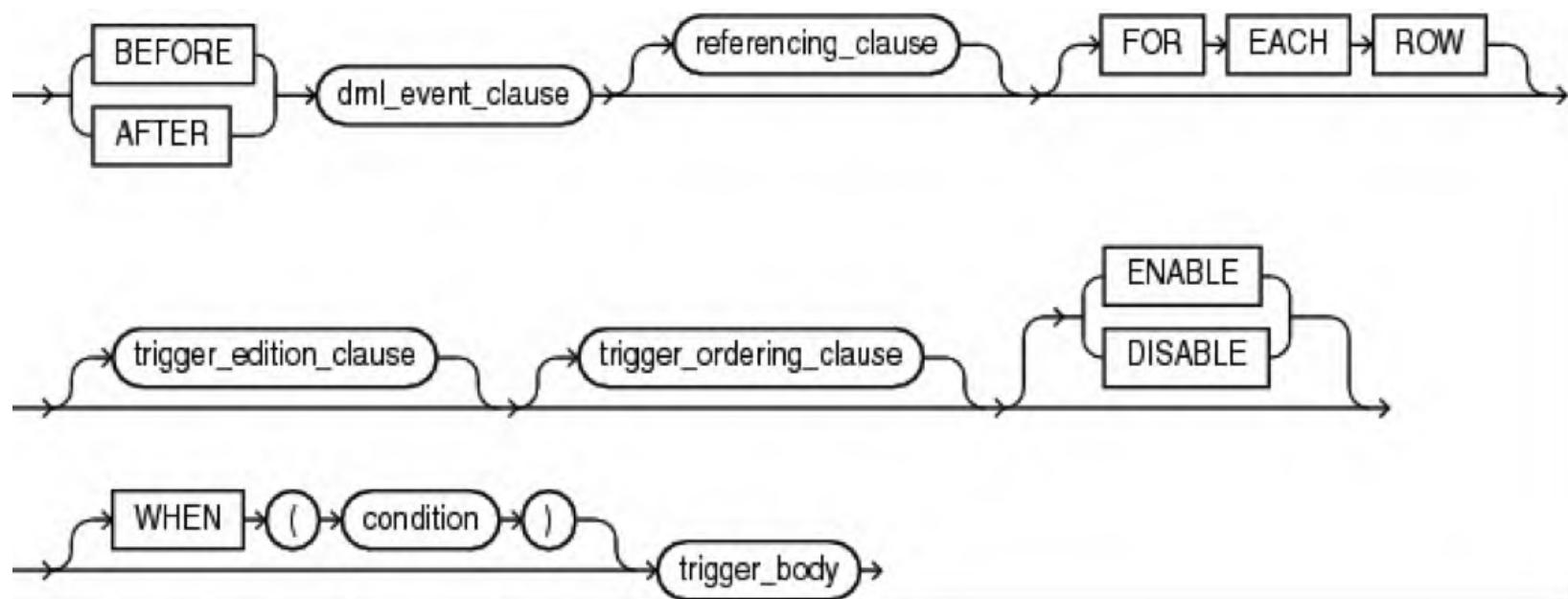
# Выражение REFERENCING

- ▶ **REFERENCING** позволяет определить имена для триггерных записей, отличные от имен по умолчанию



# Выражение WHEN

- ▶ Выражение **WHEN** определяет условия, при которых срабатывает триггер
- ▶ Хранимые функции и объектные методы не разрешены для использования в выражении WHEN



# Мутация таблиц

```
create or replace trigger tr_salary_ch
before update of salary
on employees
for each row
when (
    :new.name = 'Ivanov' and :old.salary < :new.salary
)
begin
    update employees set salary = salary *1.1
    where name <>'Ivanov';
end;

update employees set salary = 2500
where name = 'Ivanov';
```

	NAME	SALARY
1	Ivanov	2000
2	Petrov	1000
3	Smirnov	1200
4	Azimof	1300

Error starting at line 28 in command:  
update employees set salary = 2500  
where name = 'Ivanov'  
Error report:  
SQL Error: ORA-04091: таблица SYSTEM.EMPLOYEES изменяется, триггер/функция может не заметить это  
ORA-06512: на "SYSTEM.TR\_SALARY\_CH", line 2  
ORA-04088: ошибка во время выполнения триггера 'SYSTEM.TR\_SALARY\_CH'  
ORA-04091. 00000 - "table %s.%s is mutating, trigger/function may not see it"  
\*Cause: A trigger (or a user defined plsql function that is referenced in  
this statement) attempted to look at (or modify) a table that was  
in the middle of being modified by the statement which fired it.  
\*Action: Rewrite the trigger (or function) so it does not read that table.

# Мутация таблиц

```
create or replace trigger tr_salary_change
for update of salary on employees
referencing old as previous

compound trigger

    change_salary boolean;

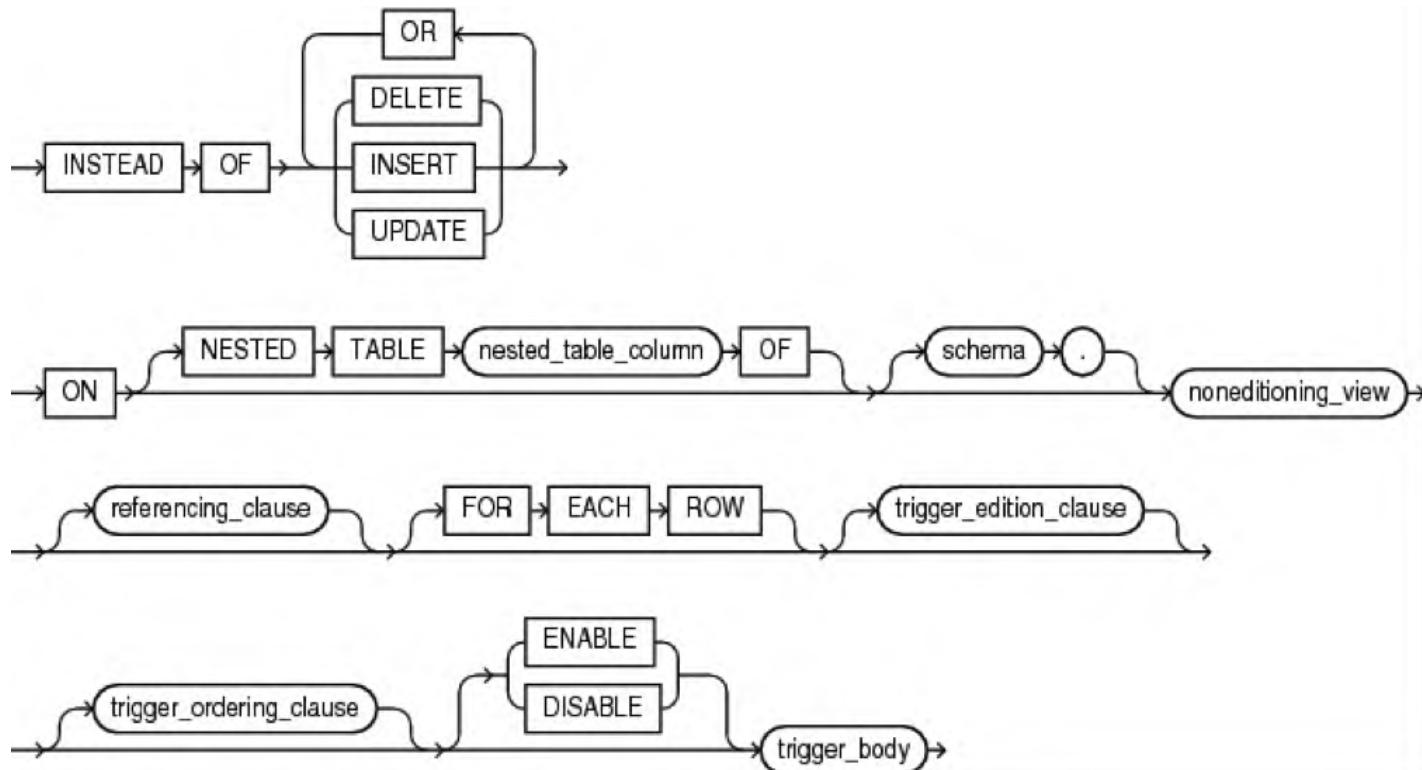
    before each row
    is
    begin
        if :new.name = 'Ivanov' and :previous.salary < :new.salary
        then
            change_salary := true;
        end if;
    end before each row;

    after statement
    is
    begin
        if change_salary then update employees
            set salary = salary *1.1 where name <>'Ivanov';
        end if;
    end after statement;
end tr_salary_change;
```

	NAME	SALARY
1	Ivanov	2500
2	Petrov	1100
3	Smirnov	1320
4	Azimof	1430

# Триггеры замещения - INSTEAD OF

- ▶ Создаются только для представлений, для таблиц нельзя.
- ▶ Только уровня строки.



# Триггеры замещения - INSTEAD OF

```
create view vauditorium
as select auditorium ua , auditorium_name na , auditorium_capacity ca, auditorium_type ta
      from AUDITORIUM

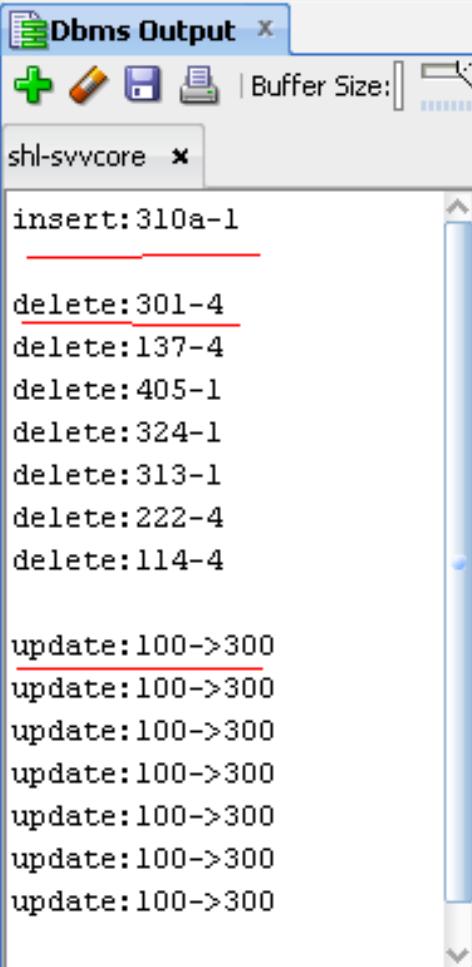
select * from vauditorium

create or replace trigger trauditatorium
instead of insert or update or delete on vAUDITORIUM
for each row
begin
  if inserting then dbms_output.put_line('insert'||:new.ua);
  elsif updating then dbms_output.put_line('update'||rtrim(:old.ca) ||'->'||:new.ca);
  elsif deleting then dbms_output.put_line('delete'||:old.ua);
  end if;
end trauditatorium;

insert into vauditorium (ua, ca)values ('310a-1',16);
delete vauditorium;
update vauditorium set ca = 300;
```



# Триггеры замещения - INSTEAD OF



The screenshot shows the Oracle SQL Developer Dbms Output window. The title bar says "Dbms Output". Below it is a toolbar with icons for new connection, edit, save, and print, followed by a "Buffer Size:" dropdown set to 1000. The main area displays the following audit log entries:

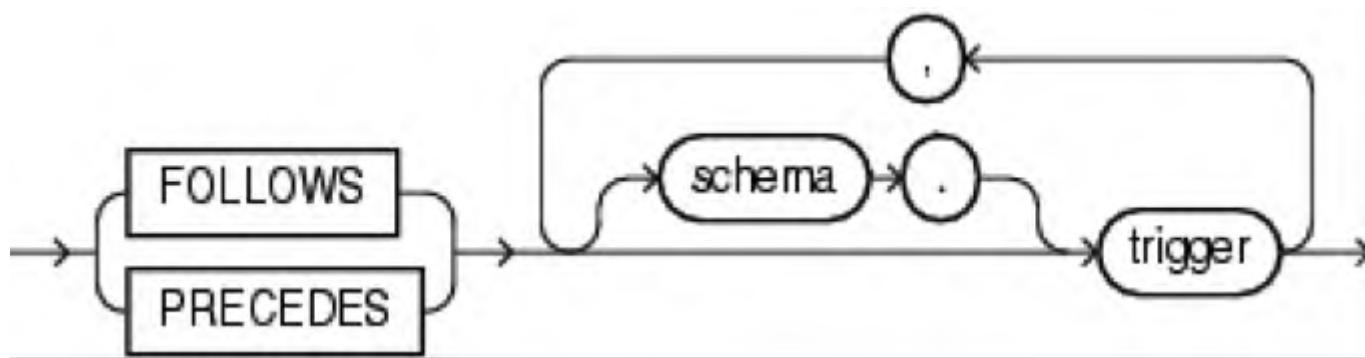
```
insert:310a-1
delete:301-4
delete:137-4
delete:405-1
delete:324-1
delete:313-1
delete:222-4
delete:114-4

update:100->300
update:100->300
update:100->300
update:100->300
update:100->300
update:100->300
update:100->300
```



# Порядок выполнения триггеров

- ▶ В каком порядке выполняются триггеры?



# Включение/отключение триггеров

---

- ▶ Включение и отключение триггеров:
  - ▶ alter trigger { disable | enable }
- ▶ Всех для таблицы:
  - ▶ ALTER TABLE table\_name { ENABLE | DISABLE } ALL TRIGGERS;
- ▶ Компиляция триггера:  
    alter trigger TRIGGER\_NAME compile;
- ▶ Переименование триггера



# Триггеры - словарь

---

- ▶ dba\_triggers
- ▶ dba\_source
- ▶ dba\_objects



# Триггеры - словарь

```
select * from user_triggers;
```

TRIGGER_NAME	trigger_type	triggering_event	table_owner	base_c
AUDITORIUM_BEFORE_ROW_2	BEFORE EACH ROW	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_BEFORE_2	BEFORE STATEMENT	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_BEFORE_ROW_1	BEFORE EACH ROW	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_BEFORE_1	BEFORE STATEMENT	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_AFTER_2	AFTER STATEMENT	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_AFTER_ROW_2	AFTER EACH ROW	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_AFTER_1	AFTER STATEMENT	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE
AUDITORIUM_AFTER ROW 1	AFTER EACH ROW	INSERT OR UPDATE OR DELETE	SVVCORE	TABLE

# Системные триггеры

---

- ▶ По времени срабатывания:
  - ▶ BEFORE, AFTER
- ▶ По уровню триггера:
  - ▶ SCHEMA, DATABASE
- ▶ По виду события:
  - ▶ 1) серверные события
  - ▶ 2) DDL-события
  - ▶ 3) события сбора статистики
  - ▶ 4) события аудита
  - ▶ 5) DCL-события



# Триггерные события DDL уровня схемы

CREATE	Событие возникает, когда создается объект базы данных
ALTER	Событие возникает, когда выполняется команда ALTER над объектом базы данных
DROP	Событие возникает, когда выполняется команда удаления объекта DROP

- ▶ К объектам события относятся таблицы, пакеты и другие объекты базы данных, которые можно найти в системном представлении ALL\_OBJECTS
- ▶ Может применяться к отдельной схеме или базе данных в целом



# Триггерные события DDL уровня схемы

```
create or replace trigger tr_drop_table
before drop on SCHEMA
begin
raise_application_error(-20000, 'Do not drop table .'||ORA_DICT_OBJ_TYPE||'.'||ORA_DICT_OBJ_NAME);
end;
```

```
drop table employees;
```

ORA-20000: Do not drop table .TABLE EMPLOYEES  
ORA-20000: Do not drop table .TABLE EMPLOYEES

```
truncate table employees;
```

table EMPLOYEES truncated.

# Триггерные события базы данных

BEFORE GRANT	При выполнении команды grant
AFTER GRANT	
BEFORE REVOKE	При выполнении команды revoke
AFTER REVOKE	
BEFORE DDL	Срабатывает на большинство команд DDL, кроме: alter database, create control file, create database
AFTER DDL	
BEFORE TRUNCATE	При выполнении команды truncate
AFTER TRUNCATE	



# Триггерные события базы данных

SERVERERROR	Событие возникает, когда сервер генерирует ошибку. Допустимы только AFTER триггеры.
LOGON	Событие возникает, когда создается соединение пользователя с базой данных. Допустимы только AFTER триггеры.
LOGOFF	Событие возникает, когда завершается соединение пользователя с базой данных. Допустимы только BEFORE триггеры.
STARTUP	Событие возникает, когда база данных открывается для работы. Допустимы только AFTER триггеры.
SHUTDOWN	Событие возникает, когда база данных закрывается. Допустимы только BEFORE триггеры.



# Системные триггеры

---

- ▶ Все кроме LOGOFF работают в режиме автофиксации
- ▶ LOGOFF входит в транзакцию отключения
- ▶ Системный триггер может генерировать исключение RAISE



# logon/logoff – триггер

```
create or replace trigger svvguest.trlogon
  after logon on svvguest.schema
begin
  insert into svvguest.eventreg(reguser, regcomment, regdate)
    values (user,'logon', sysdate);
-- dbms_output.put_line('svvguest.trlogon');
end trlogon;
```

```
create or replace trigger svvguest.trlogoff
before logoff on svvguest.schema
begin
  insert into svvguest.eventreg(reguser, regcomment, regdate)
    values (user,'logoff', sysdate);
-- dbms_output.put_line('svvguest.trlogoff');
end trlogoff;
```



# logon/logoff – триггер

select \* from SVVGUEST.eventreg

REGUSER	REGCOMMENT	REGDATE
SVVGUEST	logon	10.03.11
SVVGUEST	logon	10.03.11
SVVGUEST	logon	10.03.11
SVVGUEST	logoff	10.03.11
SVVGUEST	logon	10.03.11
SVVGUEST	logoff	10.03.11
SVVGUEST	logoff	10.03.11
SVVGUEST	logon	10.03.11
~		



# Вопросы?

---



# Секционирование данных в СУБД ORACLE

# Секционирование

Метод, позволяющий хранить сегмент данных, такой как таблица, в виде нескольких сегментов, сохраняя логическую монолитную структуру



# Информация о секционировании

*USER, ALL и DBA представления*

- ...\_TABLES – информация о таблицах
- ...\_PART\_TABLES - информация о секционированных таблицах
- ...\_TAB\_PARTITIONS - информация о табличных секциях
- ...\_PART\_KEY\_COLUMNS – информация о ключах секционирования
- ...\_SEGMENTS - информация о сегментах хранения
- ...\_OBJECTS - информация о объектах БД

# Секция – отдельный сегмент

- Разные сегменты могут находиться в разных ТП , а значит на разных дисках
- Различные секции-сегменты при общности логической структуры могут иметь собственные физические атрибуты
- Отдельные ТП находиться в состоянии OFFLINE, не нарушая работоспособности всей таблицы

# Физические атрибуты секций

```
CREATE TABLE DEACore.sales
(
    prod_id NUMBER,
    time_id DATE
)
PARTITION BY RANGE (time_id)
(
    PARTITION sales_11q1 VALUES LESS THAN ('01-APR-2011'),
    PARTITION sales_11q2 VALUES LESS THAN ('01-JUL-2011'),
    PARTITION sales_11q3 VALUES LESS THAN ('01-OCT-2011'),
    PARTITION sales_11q4 VALUES LESS THAN ('01-JAN-2012')
);
```

## *Логические и физические атрибуты*

- Каждая секция таблицы имеет общие логические атрибуты
- Имена и количество столбцов, их порядок и типы, ограничения целостности, атрибут ROW MOVEMENT
- Но каждая из секций имеет собственные физические атрибуты ТП, блочные параметры PCTFREE и PCTUSED, атрибуты COMPRESS и NOLOGGING

# *Ссылка на секцию*

*FROM PARTITION*

- Данные могут быть выбраны непосредственно из секции

```
SELECT * FROM DEACore.sales PARTITION (sales_llql);
```

- Может быть задана ассоциативная ссылка

```
SELECT * FROM DEACore.sales PARTITION FOR ('02-FEB-2011');
```

## Изменение ключа секционирования в данных

- Изменение ключа секционирования может потребовать переноса строки в другую секцию

```
UPDATE DEACore.sales PARTITION(sales_11ql)
SET time_id = time_id+90;
ORA*14402: updating partition key column would cause a partition change
```

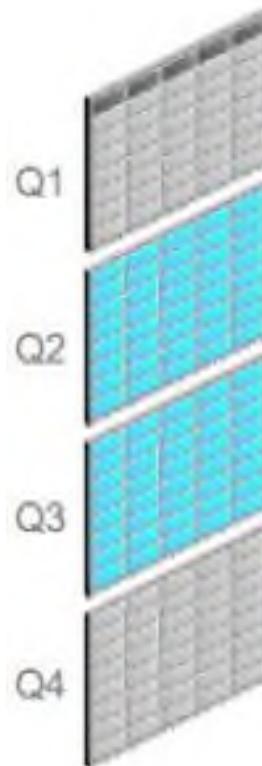
- Выполнение такого переноса нужно явным образом разрешить

```
ALTER TABLE DEACore.sales ENABLE ROW MOVEMENT;
UPDATE DEACore.sales PARTITION(sales_11ql)
SET time_id = time_id+90;
```

# Исключение секций из выборки

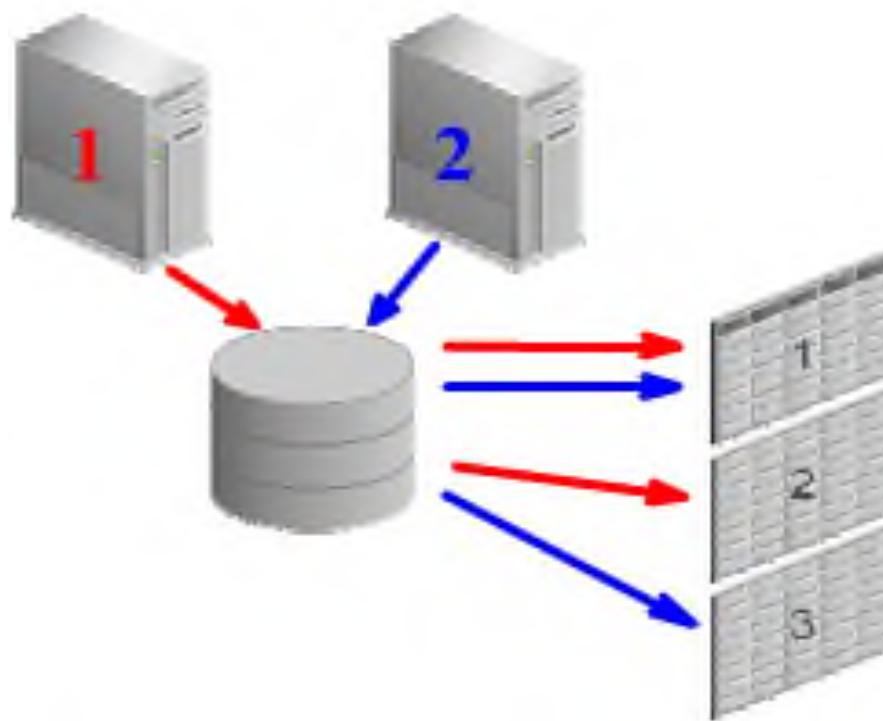
## *Partition pruning*

- Оптимизатор анализирует фразы FROM и WHERE и исключает ненужные секции из просмотра
- Ненужные секции выбираются:
  - Статически(если значение указано явно)
  - Динамически(если значение заранее не известно)



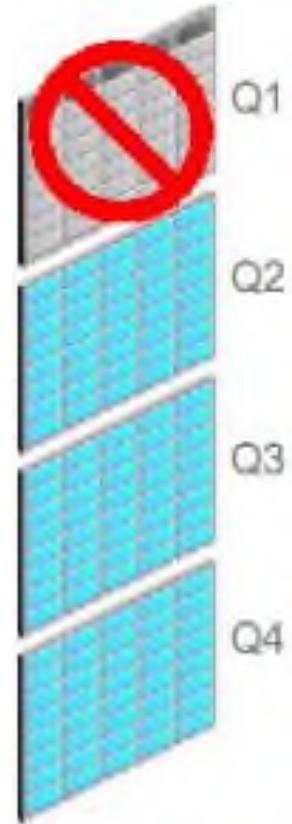
## *Параллельные операции и секционирование*

- Секция – единица параллелизма
- Секции могут сканироваться, загружаться и обновляться параллельно



# Секция в ТП *OFFLINE*

- Из-за сбоя или при обслуживании некоторые ТП могут находиться в состоянии **OFFLINE**
- При попытке чтения табличных данных из такого ТП выдаётся сообщение об ошибке  
*SELECT \* FROM DEACORE.SALES;*  
**ORA-00376: file cannot be read at this time**
- Но другие секции таблицы доступны



# Секционирование и блокировки

*Level up в иерархии блокировок*



# Операции над секциями

- *ALTER TABLE RENAME PARTITION*

```
ALTER TABLE DEACore.sales RENAME PARTITION sales_llql  
TO sales_new_name
```

- *ALTER TABLE DROP PARTITION*

```
ALTER TABLE DEACore.sales DROP PARTITION sales_llql
```

- *ALTER TABLE MOVE PARTITION*

```
ALTER TABLE DEACore.sales MOVE PARTITION sales_llql  
TABLESPACE data03 NOLOGGING;
```

- *ALTER TABLE ADD PARTITION*

```
ALTER TABLE DEACore.sales ADD PARTITION sales_max  
VALUES LESS THAN (MAXVALUE) TABLESPACE data00
```

# Операции над секциями

- *ALTER TABLE SPLIT PARTITION*

```
ALTER TABLE DEACore.sales SPLIT PARTITION sales_llq4
AT ('01-JUL-2012') INTO
(PARTITION sales_spl TABLESPACE data01,
PARTITION sales_sp2 TABLESPACE data02)
```

- *ALTER TABLE MERGE PARTITION*

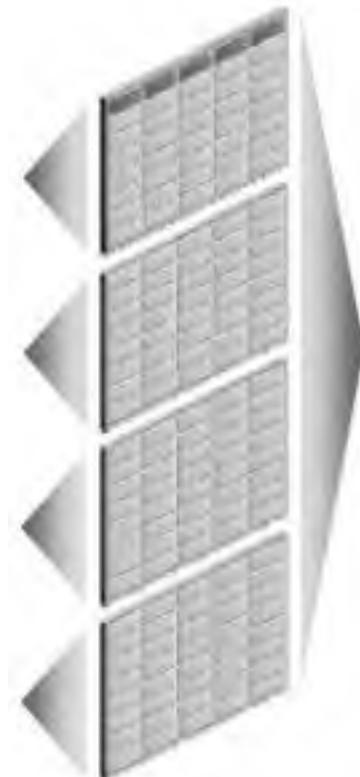
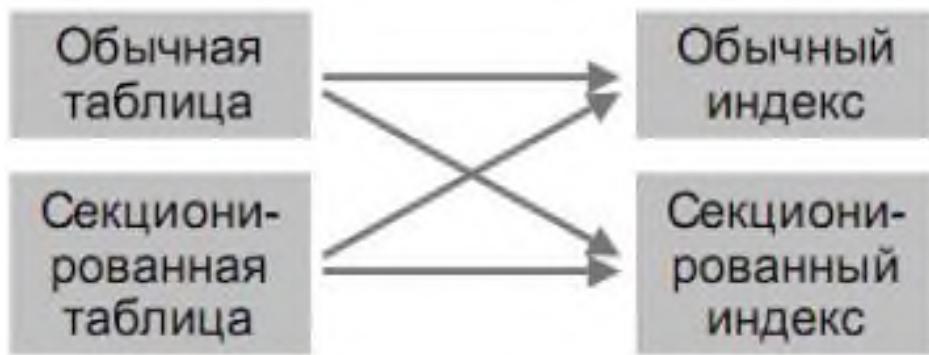
```
ALTER TABLE DEACore.sales MERGE PARTITIONS
sales_llq1, sales_llq2 INTO sales_ml;
```

- *ALTER TABLE EXCHANGE PARTITION*

```
ALTER TABLE DEACore.sales EXCHANGE PARTITION sales_llq2
WITH TABLE DEACore.my_sales WITHOUT VALIDATION;
```

# Секционирование и индексы

- Для ускорения доступа таблицы могут индексироваться
- Индексы могут секционироваться

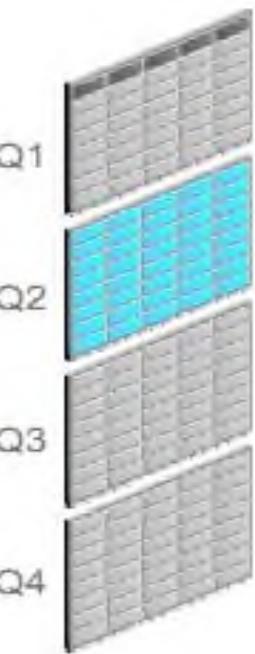


# *Локальные и глобальные индексы*

- **Локальные**(ВСЕГДА секционированные)
  - Создается автоматически для каждой секции
- 
- **Глобальные**(ИНОГДА секционированные)
  - Создаются и управляются вручную

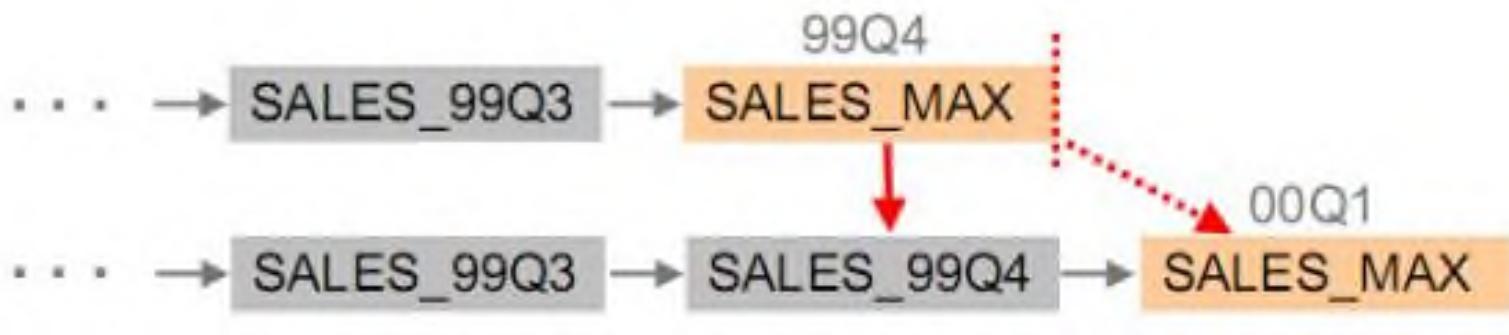
# Диапазонное (range) секционирование

```
CREATE TABLE DEACore.sales
(
    prod_id NUMBER,
    time_id DATE
)
PARTITION BY RANGE (time_id)
(
    PARTITION sales_11q1 VALUES LESS THAN ('01-APR-2011') TABLESPACE data00, Q3
    PARTITION sales_11q2 VALUES LESS THAN ('01-JUL-2011') TABLESPACE data01,
    PARTITION sales_11q3 VALUES LESS THAN ('01-OCT-2011') TABLESPACE data02,
    PARTITION sales_11q4 VALUES LESS THAN ('01-JAN-2012') TABLESPACE data03, Q4
    PARTITION sales_max VALUES LESS THAN (maxvalue) TABLESPACE data04
);
```



## *Недостаток диапазонного секционирования*

- При загрузке новых данных в таблицу нужно постоянно расщеплять секцию MAXVALUE



# Интервальное секционирование

```
create table t3 (x1 nvarchar2(50), x2 number(3))
partition by range (x2)
interval (100) store in (users)
(
    partition t11 values less than (101)
);
```

- Создается единственная диапазонная секция без MAXVALUE
- Новые секции будут создаваться **автоматически** (по 1ой операции INSERT, не попадающей в диапазоны существующих секций)

# Интервальное секционирование

```
select * from user_TAB_PARTITIONS where table_name = 'T3';
```

Script Output × Query Result ×

SQL | All Rows Fetched: 2 in 0,047 seconds

	TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE
1	T3	NO	T11		0 101
2	T3	NO	SYS_P471		0 801

```
select * from user_TAB_PARTITIONS where table_name = 'T3';
```

Script Output × Query Result ×

SQL | All Rows Fetched: 4 in 0,032 seconds

	TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE
1	T3	NO	T11		0 101
2	T3	NO	SYS_P472		0 301
3	T3	NO	SYS_P473		0 501
4	T3	NO	SYS_P471		0 801

# Интервальное секционирование

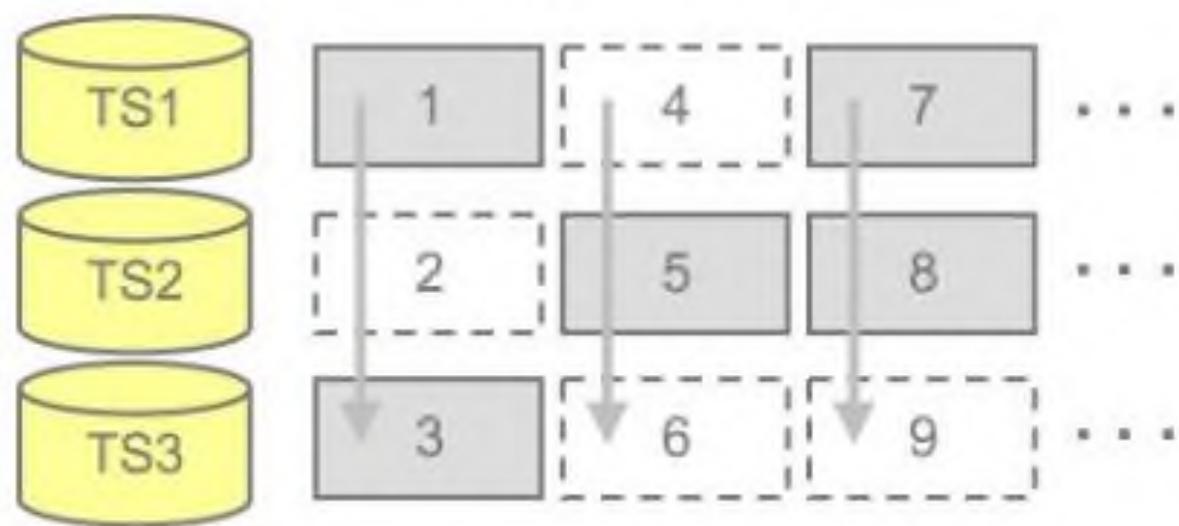
```
3 | select * from t3 partition for(100);
```

Script Output X Query Result X

SQL | All Rows Fetched: 2 in 0 seconds

	X1	X2
1	Восемь	8
2	Восемьдесят	80

# Размещение интервальных секций

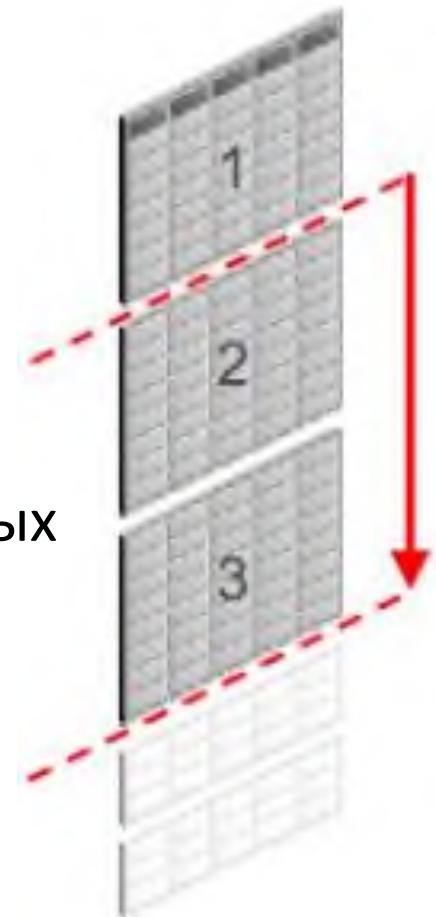


## *Изменение интервала секционирования*

- Предустановленный интервал для секций можно изменить

```
alter table t3 set interval (50);
```

- На содержимом существующих интервальных секций это никак не сказывается
- Новые секции будут создаваться для интервала нового значения



# Хэш-секционирование



# Хэш-секционирование

```
CREATE TABLE DEACore.salesH
(
    prod_id NUMBER,
    time_id DATE
)
PARTITION BY HASH (prod_id)
PARTITIONS 16
STORE IN (data00, data01, data02, data03);
```

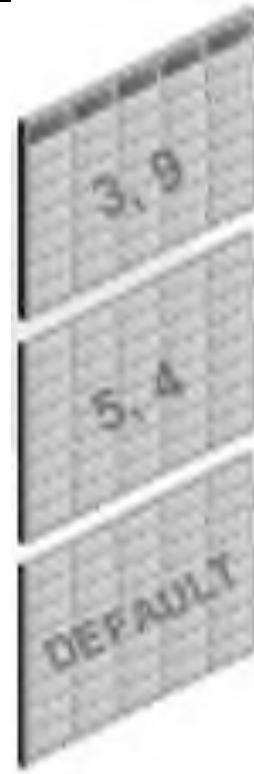


- Хэширование строк не обозначает, что строки распределяются по секциям случайнym образом

# Списочное секционирование

- Списочное секционирование позволяет разбить таблицу по списку дискретных значений

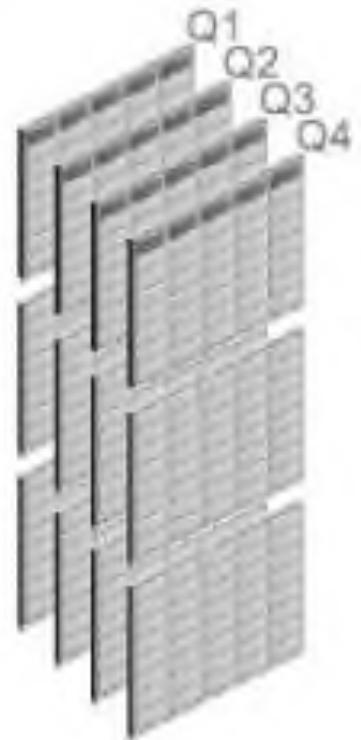
```
CREATE TABLE DEACore.salesL
(
    prod_id NUMBER,
    time_id DATE,
    group_id NUMBER
)
PARTITION BY LIST (group_id)
(
    PARTITION sales_11 VALUES (3, 9),
    PARTITION sales_12 VALUES (5, 4),
    PARTITION sales_other VALUES (DEFAULT)
);
```



# Композитное секционирование

## Секционирование секций

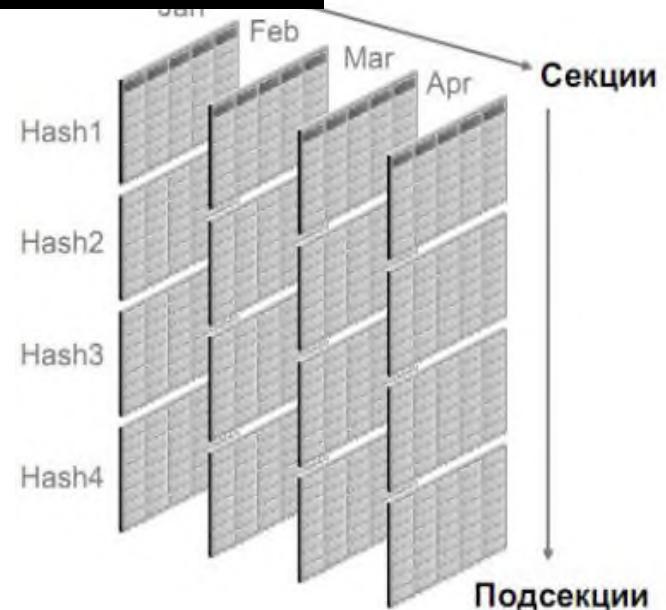
- При секционировании большой таблицы сами секции могут оказаться достаточно крупными
- Такие секции могут секционироваться по другому критерию секционирования
- Допускается 2 уровня секционирования(секции и подсекции)



# Композитное секционирование

## Секции и подсекции

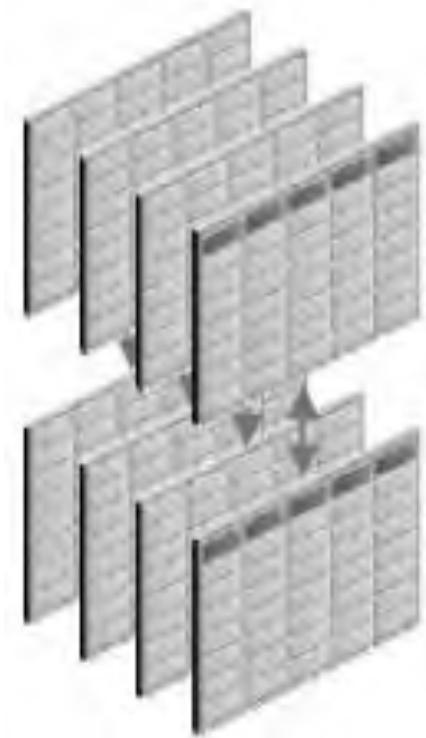
```
CREATE TABLE DEACore.salesC
(
    prod_id NUMBER,
    time_id DATE,
    cust_id NUMBER
)
PARTITION BY RANGE (time_id)
SUBPARTITION BY HASH(cust_id)
SUBPARTITIONS 4 STORE IN (data00, data01, data02, data03)
(
    PARTITION sales_llq1 VALUES LESS THAN ('01-APR-2011'),
    PARTITION sales_llq2 VALUES LESS THAN ('01-JUL-2011')
);
```



# Эквисекционирование

*Equi-partitioning*

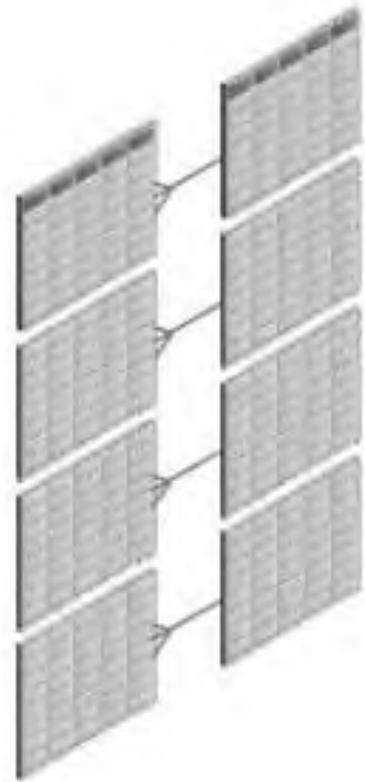
- Объекты эквисекционированы, если они секционированы одинаковым образом
- Общий метод секционирования
- Равное количество секций
- Те же границы, количество, порядок и типы столбцов в ключе
- Объекты могут быть эквисекционированы на одном или разных уровнях(секции, подсекции)



# Секционирование по ссылке

## *Reference partitioning*

- Выполняется если таблицы явным образом связаны ссылочными ограничениями целостности
- Метод и ключ секционирования подчиненной таблицы наследуется из главной
- Манипуляции с секциями главной таблицы автоматически отражаются на секциях подчиненной



# Пример ссылочного секционирования

```
CREATE TABLE DEACore.salesR
(
    prod_id NUMBER,
    time_id DATE,
    fact_id NUMBER CONSTRAINT sales_pk PRIMARY KEY
)
PARTITION BY RANGE (time_id)
(
    PARTITION sales_llq1 VALUES LESS THAN ('01-APR-2011'),
    PARTITION sales_llq2 VALUES LESS THAN ('01-JUL-2011'),
    PARTITION sales_max VALUES LESS THAN (MAXVALUE)
);

CREATE TABLE DEACore.salesR_details
(
    details_id NUMBER,
    fact_id NUMBER NOT NULL
    CONSTRAINT sales_details_fk REFERENCES DEACore.salesR(fact_id)
)
PARTITION BY REFERENCE (sales_details_fk);
```

- Родительская таблица не может быть интервальной

# *Виртуальные столбцы в таблице*

- Значения в столбцах вычисляются динамически(“на лету”)

```
CREATE TABLE DEACore.salesR
(
    prod_id NUMBER,
    AverageRevenue NUMBER(10, 2),
    TotalProduct NUMBER(10, 2),
    TotalRevenue IS (AverageRevenue*TotalProduct)
);
```



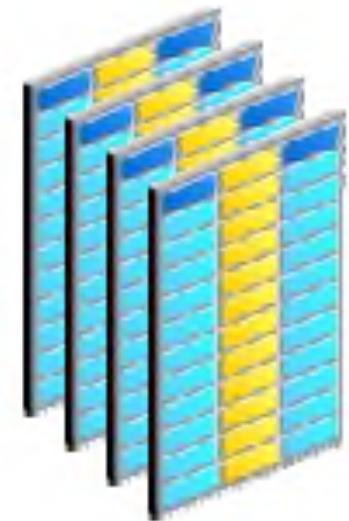
- Виртуальными могут быть только столбцы в обычных таблицах

## *Описание виртуального столбца*

- Выражение должно быть детерминистическим (независимым от времени)
- Выражение может ссылаться только на другие столбцы из этой же таблицы(но не виртуальные)

# Секционирование по виртуальному столбцу

```
CREATE TABLE DEACore.salesR
(
    prod_id NUMBER,
    AverageRevenue NUMBER(10, 2),
    TotalProduct NUMBER(10, 2),
    TotalRevenue AS (AverageRevenue*TotalProduct)
)
PARTITION BY RANGE (TotalRevenue)
(
    PARTITION TR1 VALUES LESS THAN (1000) TABLESPACE data00,
    PARTITION TR2 VALUES LESS THAN (5000) TABLESPACE data01,
    PARTITION TR3 VALUES LESS THAN (10000) TABLESPACE data02,
    PARTITION TR_max VALUES LESS THAN (maxvalue) TABLESPACE data03
);
```



- Виртуальный столбец может быть использован практически в любых схемах секционирования

# Зачем секционировать данные?

- Секционирование повышает производительность обработки данных
- Отбрасывание секций, параллелизм, балансировка дисковой нагрузки
- Секционирование упрощает управление крупными объектами хранения
- Крупный объект разбивается на индивидуально-управляемые части
- Секционирование обеспечивает дополнительную надежность системы
- Сокращение времени обслуживания, нечувствительность к сбоям

# **ORACLE 12с**

**Системные пакеты**

**Лекция 15**

# Системные пакеты Oracle

- ▶ Встроенные системные пакеты:
- ▶ Устанавливаются во время установки Oracle
- ▶ Используются для расширения основных функциональных возможностей Oracle
- ▶ Владелец пакетов - SYS (обычно, есть и другие)
- ▶ Написаны на С или на PL/SQL

```
SELECT COUNT(*) FROM DBA_Objects_AE where Object_Type = 'PACKAGE';
```

Query Result x

SQL | All Rows Fetched: 1 in 0,047 seconds

	COUNT(*)
1	1297

# Системные пакеты Oracle

```
SELECT Object_Type, OWNER, Object_Name FROM DBA_Objects_AE
WHERE
Object_Type = 'PACKAGE'
ORDER BY Owner, Object_Name;
```

683	PACKAGE	SYS	DBMS_FBT
684	PACKAGE	SYS	DBMS_FEATURE_USAGE
685	PACKAGE	SYS	DBMS_FEATURE_USAGE_INTERNAL
686	PACKAGE	SYS	DBMS_FEATURE_USAGE_REPORT
687	PACKAGE	SYS	DBMS_FGA
688	PACKAGE	SYS	DBMS_FILE_GROUP
689	PACKAGE	SYS	DBMS_FILE_GROUP_DECL
690	PACKAGE	SYS	DBMS_FILE_GROUP_EXP
691	PACKAGE	SYS	DBMS_FILE_GROUP_EXP_INTERNAL
692	PACKAGE	SYS	DBMS_FILE_GROUP_IMP
693	PACKAGE	SYS	DBMS_FILE_GROUP_IMP_INTERNAL
694	PACKAGE	SYS	DBMS_FILE_GROUP_INTERNAL_INVOK
695	PACKAGE	SYS	DBMS_FILE_GROUP_UTL
696	PACKAGE	SYS	DBMS_FILE_GROUP_UTL_INVOK
697	PACKAGE	SYS	DBMS_FILE_TRANSFER
698	PACKAGE	SYS	DBMS_FLASHBACK
699	PACKAGE	SYS	DBMS_FLASHBACK_ARCHIVE

# Пакеты APEX

---

- ▶ APEX – Oracle Application Express – среда разработки веб-приложения
- ▶ APEX\_CUSTOM\_AUTH – проверка подлинности и управления сеансом
- ▶ APEX\_APPLICATION – использование глобальных переменных
- ▶ APEX\_ITEM – создание элементов форм на основе SQL-запроса
- ▶ APEX\_UTIL – различные утилиты состояния сеанса, файлов, авторизации и пр.



## Пакеты dbms\_

```
■ SELECT Object_Type, OWNER, Object_Name  
      FROM DBA_Objects_AE  
        WHERE Object_Type = 'PACKAGE'  
          AND Object_Name LIKE 'DBMS_%'  
 ORDER BY Owner, Object_Name;
```

587	PACKAGE	XDB	DBMS_XMLSCHEMA_INT	
588	PACKAGE	XDB	DBMS_XMLSCHEMA LSB	
589	PACKAGE	XDB	DBMS_XMLSTORAGE_MANAGE	
590	PACKAGE	XDB	DBMS_XMLTRANSLATIONS	
591	PACKAGE	XDB	DBMS_XSLPROCESSOR	

# DBMS\_ADVANCED\_REWRITE

- ▶ DBMS\_ADVANCED\_REWRITE – перехват и замена SQL-запросов

```
3 CREATE TABLE ST(NAME_ST NVARCHAR2(25), SURNAME_ST NVARCHAR2(25));
4
5 INSERT INTO ST VALUES ('FIRST', 'STUDENT');
6 INSERT INTO ST VALUES ('SECOND', 'STUDENT');
7 COMMIT;
8
9 BEGIN
10   sys.dbms_advanced_rewrite.declare_rewrite_equivalence(
11     'query',
12     'select NAME_ST from ST',
13     'select SURNAME_ST from ST',
14     false);
15 END;
16
17 ALTER SESSION SET query_rewrite_integrity = TRUSTED;
18
19 SELECT NAME_ST from ST;
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0,016 seconds

NAME_ST
1 STUDENT
2 STUDENT

## DBMS ADVISED

---

- ▶ DBMS ADVISED – часть набора экспертной системы для решения проблем производительности, связанных с компонентами сервера базы данных
- ▶ Требуется grant role ADVISOR
- ▶ Используется в Enterprise Manager и Automatic Database Diagnostic Monitor (ADDM)
  
- ▶ DBMS\_SQLTUNE – сбор статистики, используется при анализе производительности SQL-операторов



# DBMS\_APPLICATION\_INFO

- ▶ DBMS\_APPLICATION\_INFO – присвоение имени процессу для удобства мониторинга и отладки

```
-- DBMS_application_info
exec dbms_application_info.set_client_info('VIEWABLE PROGRAM');

SELECT client_info  FROM v$session WHERE  client_info IS NOT NULL;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab is active, indicated by a green play button icon. Below the tabs, there are icons for 'Script', 'SQL', and 'X'. A status message 'All Rows Fetched: 1 in 0 seconds' is displayed. The main area shows the query results:

CLIENT_INFO
1 VIEWABLE PROGRAM

# DBMS\_AQ

---

- ▶ DBMS\_AQ – пакет для Advanced Queuing – система обмена очередями сообщений
- ▶ DBMS\_TRANSFORM – пакет для Oracle Advanced Queuing – преобразования данных
- ▶ Процессы:
  - ▶ Процесс захвата изменений - Capture Process
  - ▶ Процесс передачи изменений - Propagation Process
  - ▶ Процесс внесения изменений - Apply Process
  - ▶ Очередь - queue



## DBMS\_DEFER

---

- ▶ DBMS\_DEFER – вызов удаленных процедур
- ▶ DBMS\_OFFLINE\_OG, DBMS\_REPCAT – поддержка репликации
- ▶ DBMS\_REFRESH – создание группы материализованных представлений для обновления как единого целого
- ▶ Среди прочих пакетов используются Oracle Advanced Replication



## DBMS\_BACKUP\_RESTORE

---

- ▶ DBMS\_BACKUP\_RESTORE – пакет для создания резервных копий для управляющих файлов, журналов повтора, архивных файлов



## DBMS\_CRYPTO

---

- ▶ DBMS\_CRYPTO – пакет шифрования данных
- ▶ Поддерживаются алгоритмы шифрования - DES, AES, RC4, 3DES, 3DES-2KEY



## DBMS\_CUBE, DBMS\_DIMENSION

---

- ▶ DBMS\_CUBE - поддерживает развертывание кубических материализованных представлений из существующих реляционных материализованных представлений
- ▶ DBMS\_DIMENSION - отображение информации об измерениях – иерархических связях между данными
- ▶ Data Warehouse - предметно-ориентированная база данных, предназначенная для отчетов и/или OLAP



## DBMS\_DEBUG

---

- ▶ DBMS\_DEBUG – пакет для отладки PL/SQL-кода на сервере



# DBMS\_DDL

---

- ▶ DBMS\_DDL - эквиваленты команд DDL, которые нельзя использовать в PL/SQL непосредственно:
  - ▶ ALTER COMPILE
  - ▶ CREATE\_WWRAPPED



## DBMS\_DESCRIBE

---

- ▶ DBMS\_DESCRIBE – позволяет получить информацию о процедурах



## DBMS\_FGA

---

- ▶ DBMS\_FGA – пакет настройки и использования политик аудита:
  - ▶ добавление /удаление
  - ▶ включение/отключение
- ▶ FGA - Fine-Grained Audit



# DBMS\_FILE\_TRANSFER

- ▶ DBMS\_FILE\_TRANSFER – пересылка файлов между файловыми системами и базами данных
  - ▶ COPY\_FILE
  - ▶ GET\_FILE
  - ▶ PUT\_FILE

```
|  
CREATE DIRECTORY SourceDir_1 as 'c:\DIRO1\ORCL\';  
CREATE DIRECTORY Dest_Dir_1 as 'c:\DIRO1\SRV\';  
  
BEGIN  
    DBMS_FILE_TRANSFER.COPY_FILE(  
        source_directory_object      => 'SourceDir_1',  
        source_file_name            => 'CONTROL01.CTL',  
        destination_directory_object => 'Dest_Dir_1',  
        destination_file_name       => 'CONTROL01.CTL');  
END;
```

# DMS\_FLASHBACK

---

- ▶ DMS\_FLASHBACK – пакет поддержки технологии flashback:
  - ▶ DISABLE
  - ▶ ENABLE\_AT\_SYSTEM\_CHANGE\_NUMBER
  - ▶ ENABLE\_AT\_TIME
  - ▶ GET\_SYSTEM\_CHANGE\_NUMBER
  - ▶ TRANSACTION\_BACKOUT



## DMS\_ HS\_PASSTHROUGH

---

- ▶ HS – Heterogeneous Services
- ▶ Пакет поддержки интерфейса с базами данных  
отличными от Oracle – ANSI SQL для поддерживаемых  
баз данных



## DBMS\_JAVA

---

- ▶ DBMS\_JAVA – пакет работы с хранимыми Java-процедурами



## DMBS\_LDAP

---

- ▶ LDAP – Lightweight Directory Access Protocol
- ▶ DMBS\_LDAP – доступ к LDAP-серверам



## DBMS\_JOB

---

- ▶ DBMS\_JOB – поддержка управления заданиями
- ▶ Задание – процедура, PL-SQL блок, внешняя процедура
- ▶ Выполняется в фоновом режиме, надо задать количество одновременно выполняемых процессов

```
ALTER SYSTEM SET JOB_QUEUE_PROCESSES=9;
```



## DBMS\_JOB

---

- ▶ SUBMIT – создание задания
  - ▶ ISUBMIT – создание задания с номером
  - ▶ REMOVE – удаление задания из очереди
  - ▶ RUN – немедленное выполнение задания в пользовательском сеансе
  - ▶ BROKEN – разрушение задания (16)
  - ▶ INSTANCE – указание экземпляра
  - ▶ NEXT\_DATE – изменение времени выполнения
  - ▶ INTERVAL – изменение интервала выполнения
  - ▶ CHANGE – изменение параметров задания
  - ▶ WHAT – изменение задания
-

# DBMS\_JOB

```
CREATE OR REPLACE PROCEDURE up_students_grades_JOB IS
BEGIN
    UPDATE STUDENT_GRADES SET GRADE= GRADE*1.1 WHERE IDSTUDENT = 123;
    UPDATE STUDENT_GRADES SET GRADE= GRADE*0.9 WHERE IDSTUDENT = 125;
END;

declare job_number user_jobs.job%type;
begin
dbms_job.submit(job_number, 'begin up_students_grades_JOB; end;',
    SYSDATE, 'SYSDATE + 60/86400'); -- 24 * 60 * 60 = 86 400
COMMIT;
SYS.dbms_output.put_line(job_number);
end;
```

41

system x

```
select job, what, last_date, last_sec, next_date, next_sec, broken from user_jobs where job=41;
```

	JOB	WHAT	LAST_DATE	LAST_SEC	NEXT_DATE	NEXT_SEC	BROKEN
1	41	begin up_students_grades_JOB; end;	12.12.17	17:42:24	12.12.17	17:43:24	N

# DBMS\_JOB

```
begin
    dbms_job.remove(41);
end;

--isubmit
declare job_number user_jobs.job%type;
begin
dbms_job.isubmit(43, 'begin up_students_grades_JOB; end;',
    SYSDATE, 'SYSDATE + 60/86400'); -- 24 * 60 * 60 = 86 400
COMMIT;
select job into job_number from user_jobs;
SYS.dbms_output.put_line(job_number); --43
end;

-- run
declare job_number user_jobs.job%type;
begin
dbms_job.run(43);
COMMIT;
select job into job_number from user_jobs;
SYS.dbms_output.put_line(job_number); --43
end;
```



# DBMS\_JOB

```
-- change
declare job_number user_jobs.job%type;
begin
dbms_job.change(43, -- номер задания
                 null, --what
                 null, -- следующее исполнение
                 'SYSDATE + 360/86400'); --интервал
COMMIT;
end;
```

```
-- next_date -- interval
begin
dbms_job.next_date(43,SYSDATE + 2/24); -- через 2 часа
dbms_job.interval(43,SYSDATE + 3); -- через 31 дня
COMMIT;
end;
```

```
-- broken
begin
dbms_job.broken(43,true, null);
end;
```



# DBMS\_SCHEDULER

---

- ▶ Schedule – расписание
- ▶ Program - программа
- ▶ Job – плановая программа



# DBMS\_SCHEDULER

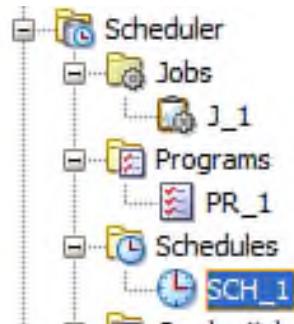
```
--scheduler
begin
dbms_scheduler.create_schedule(
    schedule_name =>'Sch_1',
    start_date =>'01/01/2017 12:00:00',
    repeat_interval =>'FREQ=DAILY',
    comments => 'Sch_1 DAILY at 12:00');
|
end;

---program
begin
dbms_scheduler.create_program(
    program_name =>'Pr_1',
    program_type =>'STORED PROCEDURE',
    program_action =>'up_students_grades_JOB',
    number_of_arguments =>0,
    enabled => false,
    comments => 'up_students_grades_JOB');
end;

---job
begin
dbms_scheduler.create_job(
    job_name =>'J_1',
    program_name =>'Pr_1',
    schedule_name =>'Sch_1',
    enabled => true);
end;
```



# DBMS\_SCHEDULER



Name	Value
1 SCHEDULE_NAME	SCH_1
2 SCHEDULE_TYPE	CALENDAR
3 START_DATE	01.01.17 12:00:00,000000000
4 REPEAT_INTERVAL	FREQ=DAILY
5 EVENT_QUEUE_OWNER	(null)
6 EVENT_QUEUE_NAME	(null)
7 EVENT_QUEUE_AGENT	(null)
8 EVENT_CONDITION	(null)
9 FILE_WATCHER_OWNER	(null)
10 FILE_WATCHER_NAME	(null)
11 END_DATE	(null)
12 COMMENTS	Sch_1 DAILY at 12:00

Name	Value
1 PROGRAM_NAME	PR_1
2 PROGRAM_TYPE	STORED_PROCEDURE
3 PROGRAM_ACTION	up_students_grades_JOB
4 NUMBER_OF_ARGUMENTS	0
5 ENABLED	FALSE
6 COMMENTS	up_students_grades_JOB

Name	Value
1 OWNER	SYSTEM
2 JOB_NAME	J_1
3 JOB_CLASS	DEFAULT_JOB_CLASS
4 COMMENTS	(null)
5 ENABLED	TRUE
6 CREDENTIAL_NAME	(null)
7 DESTINATION	(null)
8 PROGRAM_NAME	PR_1
9 JOB_TYPE	(null)
10 JOB_ACTION	(null)
11 NUMBER_OF_ARGUMENTS	(null)
12 SCHEDULE_OWNER	SYSTEM
13 SCHEDULE_NAME	SCH_1
14 START_DATE	01.01.17 12:00:00,0000000
15 REPEAT_INTERVAL	(null)
16 END_DATE	(null)
17 EVENT_QUEUE_OWNER	(null)

# DBMS\_JOB, DBMS\_SCHEDULER

---

- ▶ Представления словаря:
  - ▶ DBA\_JOBS
  - ▶ USER\_JOBS
  
- ▶ USER\_SCHEDULER\_SCHEDULES
- ▶ USER\_SCHEDULER\_PROGRAMS
- ▶ USER\_SCHEDULER\_JOBS
- ▶ USER\_SCHEDULER\_JOB\_LOG



# DBMS\_JOB, DBMS\_SCHEDULER

---

- ▶ Права:
  - ▶ Роль SCHEDULERADMIN
  - ▶ Права на объекты



# DBMS\_LOB

---

- ▶ DBMS\_LOB - работа с большими объектами – изображения, звук, видео, файлы
  - ▶ **BLOB** – неструктурированные двоичные данные
  - ▶ **CLOB** – символьные данные
  - ▶ **BFILE** – внешние файлы операционной системы, содержащие двоичные данные
- ▶ LOB - объект состоит из локатора и значения:
  - ▶ Локатор – это внутренний указатель на фактическое значение большого объекта
  - ▶ Значение – это реальное содержимое объекта
- ▶ Доступ к значениям больших объектов осуществляется с помощью соответствующего локатора



# DBMS\_LOCK

---

- ▶ DBMS\_LOCK – пакет управления блокировками:
  - ▶ Задать блокировку
  - ▶ Задать уникальное имя блокировки
  - ▶ Изменить режим блокировки
  - ▶ Снять блокировку



## DBMS\_LOGMNR

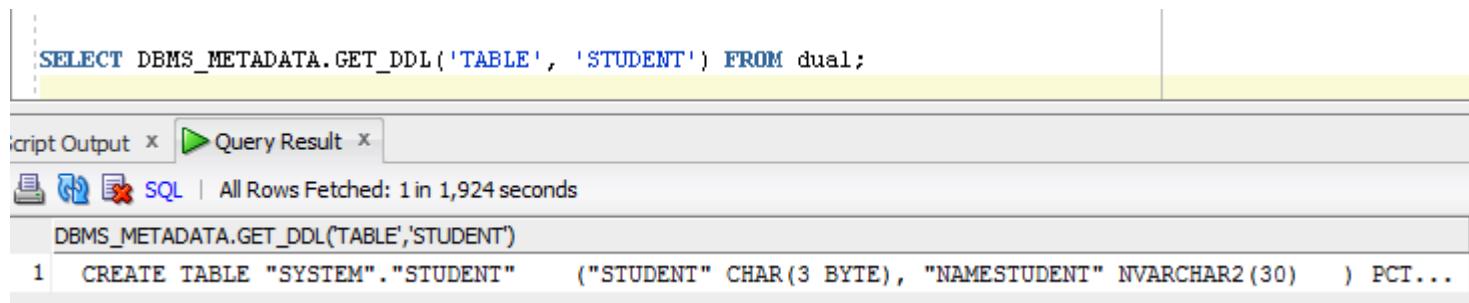
---

- ▶ DBMS\_LOGMNR – пакет поддержки технологии Log Miner – анализ данных журналов повторного выполнения



# DBMS\_METADATA

- ▶ DBMS\_METADATA – позволяет производить извлечение из словаря информации об объектах базы данных в виде XML или DDL



The screenshot shows the Oracle SQL Developer interface. In the top-left query editor, the following SQL statement is entered:

```
SELECT DBMS_METADATA.GET_DDL('TABLE', 'STUDENT') FROM dual;
```

Below the query editor, the results are displayed in the "Query Result" tab. The results show the DDL for the STUDENT table:

```
DBMS_METADATA.GET_DDL('TABLE','STUDENT')
1 CREATE TABLE "SYSTEM"."STUDENT" ("STUDENT" CHAR(3 BYTE), "NAMESTUDENT" NVARCHAR2(30) ) PCT...
```

The interface includes tabs for "Script Output" and "Query Result". The "Query Result" tab is active, showing the results of the query.

# DBMS\_MONITOR

---

- ▶ DBMS\_MONITOR – предоставление данных:
  - ▶ Сбор статистики
  - ▶ Сбор трассировочных данных



# DBMS\_OUTPUT

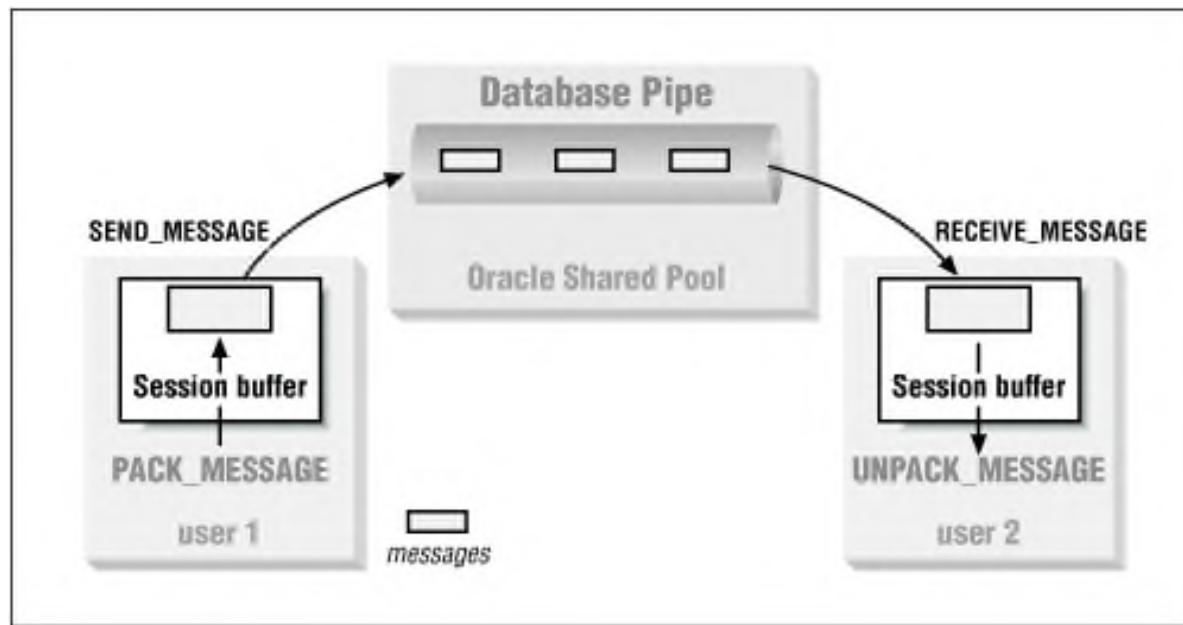
---

- ▶ DBMS\_OUTPUT – серверный вывод



# DBMS\_PIPE

- ▶ DBMS\_PIPE - обмен сообщениями между сессиями одной или нескольких баз данных



# DBMS\_PIPE

---

- ▶ CREATE\_PIPE
- ▶ NEXT\_ITEM\_TYPE
- ▶ PACK\_MESSAGE
- ▶ PURGE
- ▶ RECEIVE\_MESSAGE
- ▶ REMOVE\_PIPE
- ▶ RESET\_BUFFER
- ▶ UNIQUE\_SESSION\_NAME
- ▶ UNPACK\_MESSAGE



## DBMS\_TRACE, DBMS\_PROFILER, DBMS\_HPROF

---

- ▶ Пакеты сбора данных для анализа производительности
- ▶ Сохраняются данные о выполнении различных процедур
- ▶ Плоский и иерархический профайлеры



# DBMS\_RANDOM

- ▶ DBMS\_RANDOM - генератор случайных чисел

```
-----  
EGIN
```

```
DBMS_RANDOM.SEED (23);  
dbms_output.put_line( dbms_random.value(3,20) );  
dbms_output.put_line( dbms_random.string ('u',7));  
dbms_output.put_line( dbms_random.string ('l',6));  
dbms_output.put_line( dbms_random.string ('a',10));  
dbms_output.put_line( dbms_random.string ('x',12));  
dbms_output.put_line( dbms_random.string ('p',6));
```

```
3,98236923437162718745253897112179511632  
TNPRHVI  
zcegfb  
QScKhiiztl  
LGWL8DYPGCGW3  
t^N4$^
```

```
END;
```



# DBMS\_REDACT

- ▶ DBMS\_REDACT – пакет, который предоставляет интерфейс для работы с Oracle Data Redaction
- ▶ Маскировка данных, возвращаемых из запросов

```
CREATE TABLE CLIENT_INFO
(
    ID NUMBER,
    F_NAME VARCHAR2(64), /* фамилия*/
    NAME VARCHAR2(64), /* имя */
    S_NAME VARCHAR2(64), /* отчество */
    BIRTHDAY DATE, /* дата рождения */
    PHONE VARCHAR2(32), /* телефон */
    EMAIL VARCHAR2(64), /* email */
    CCARD VARCHAR2(32), /* номер кредитной карты */
    CONSTRAINT "CLIENT_INFO_PK" PRIMARY KEY ("ID")
);

insert into CLIENT_INFO values(10, 'Иванов', 'Иван', 'Иванович', to_date('15-05-1986', 'DD-MM-YYYY'),
'79763334589', 'ivan@dom2.ru', '5767881897856776');
```

# DBMS\_REDACT

- ▶ Создаем политику и добавляем правило, по которому день в дате рождения заменяется на первое число:

```
begin
  DBMS_REDACT.ADD_POLICY(
    object_schema      => 'GPRCORE',
    object_name        => 'CLIENT_INFO',
    column_name        => 'BIRTHDAY',
    policy_name        => 'redact_client_info',
    function_type      => DBMS_REDACT.PARTIAL, /* Частичное маскирование */
    function_parameters => 'Md01Y', /* Мaska изменений */
    expression          => 'SYS_CONTEXT(''SYS_SESSION_ROLES'', ''RL_GPRCORE1'') = ''FALSE''' /* Условие - замена при отсутствии
);
END;
```

# DBMS\_REDACT

- ▶ Добавляем в policy условие показывать только 5 последних цифр телефона:

```
BEGIN
DBMS_REDACT.ALTER_POLICY(
    object_schema      => 'GPRCORE',
    object_name        => 'CLIENT_INFO',
    column_name        => 'PHONE',
    policy_name        => 'redact_client_info',
    function_type      => DBMS_REDACT.REGEXP, /* Маскирование с помощью регулярного выражения */
    regexp_pattern     => '\d+(\d{5})$',
    regexp_replace_string => '*****\1',
    regexp_position    => DBMS_REDACT.RE_BEGINNING,
    regexp_occurrence   => DBMS_REDACT.RE_ALL,
    expression          => 'SYS_CONTEXT(''SYS_SESSION_ROLES'', ''RL_GPRCORE1'') = ''FALSE''',
    action              => DBMS_REDACT.ADD_COLUMN
);
END;
```

# DBMS\_REDACT

- ▶ Добавляем в политику условие скрывать домен электронной почты

```
BEGIN  
DBMS_REDACT.ALTER_POLICY(  
    object_schema      => 'GPRCORE',  
    object_name        => 'CLIENT_INFO',  
    column_name        => 'EMAIL',  
    policy_name        => 'redact_client_info',  
    function_type      => DBMS_REDACT.REGEXP,  
    regexp_pattern     => DBMS_REDACT.RE_PATTERN_EMAIL_ADDRESS, /* Используем готовое */  
    regexp_replace_string => DBMS_REDACT.RE_REDACT_EMAIL_DOMAIN, /* Используем готовое */  
    expression          => 'SYS_CONTEXT(''SYS_SESSION_ROLES'', ''RL_GPRCORE1'') = ''FALSE''',  
    action              => DBMS_REDACT.ADD_COLUMN  
);  
END;
```

# DBMS\_REDACT

```
select f_name, name, birthday, email, phone, ccard from client_info;
```

Query Result

SQL | All Rows Fetched: 3 in 0,986 seconds

	F_NAME	NAME	BIRTHDAY	EMAIL	PHONE	CCARD
1	Иванов	Иван	01.05.86	ivan@xxxxx.com	*****34589	576788XXXXXX6776

SQL | All Rows Fetched: 1 in 1,034 seconds

	F_NAME	NAME	BIRTHDAY	EMAIL	PHONE	CCARD
1	Иванов	Иван	15.05.86	ivan@dom2.ru	79763334589	5767881897856776

## DBMS\_REPAIR

---

- ▶ DBMS\_REPAIR – выявление проблем и восстановление при сбоях:
- ▶ Обнаруживает и сообщает о сбоях в таблице или индексе
- ▶ Отмечает блоки как поврежденные при сканировании таблиц и индексов
- ▶ Предоставляет функции для восстановления внешних ключей «сирот»



# DBMS\_RESOURCE\_MANAGER

---

- ▶ DBMS\_RESOURCE\_MANAGER
- ▶ RESOURCE\_MANAGER\_PRIVS
- ▶ администрирование базы данных
- ▶ сопровождение планов ресурсов



## DBMS\_ROWID

---

- ▶ DBMS\_ROWID – пакет для управления ROWID:
  - ▶ создание и извлечение информации о строках таблиц
  - ▶ нахождение строки в блоке



## DBMS\_SERVICE

---

- ▶ DBMS\_SERVICE – управление службами
- ▶ Используется в single instance и в RAC (Real Application Cluster)
- ▶ позволяет создавать, удалять, запускать и останавливать службы
- ▶ позволяет отключать все сеансы удаленной службы



# DBMS\_SHARED\_POOL

- ▶ DBMS\_SHARED\_POOL – пакет, позволяющий управлять разделяемым пулом SGA:
  - ▶ загрузка больших объектов
  - ▶ часто выполняемые триггеры
  - ▶ поддержка последовательностей

```
exec dbms_shared_pool.keep('sys.aw_drop_trg', 'R');

SELECT * FROM v$db_object_cache WHERE kept = 'YES' and type = 'TRIGGER';
```

Script Output x Query Result x

All Rows Fetched: 1 in 0,047 seconds

SQL

	OWNER	NAME	DB_LINK	NAMESPACE	TYPE	SHARABLE_MEM	LOADS	EXECUTIONS	LOCKS	PINS	KEPT
1	SYS	AW_DROP_TRG (null)	TRIGGER	TRIGGER		9192	1	0	0	0	YES

## DBMS\_SPACE

---

- ▶ DBMS\_SPACE – пакет управления потребностью в дисковом пространстве:
  - ▶ Наличие / отсутствие свободных блоков
  - ▶ Рост объектов



# DBMS\_SQL

- ▶ DBMS\_SQL – пакет для работы с динамическим SQL

```
CREATE PROCEDURE create_table
    (table_name IN OUT VARCHAR2)
AS
    cur INTEGER;
    rest INTEGER;
    str VARCHAR2(250);
BEGIN
    table_name := 'ST' || dbms_session.unique_session_id;
    str := 'CREATE TABLE ' || table_name
          || ' (gr INTEGER,
          || name VARCHAR2(14))';

    cur := dbms_sql.open_cursor;
    dbms_sql.parse(cur, str, dbms_sql.NATIVE);
    rest := dbms_sql.execute(cur);
    dbms_sql.close_cursor(cur);
END;
```

## DBMS\_STANDARD, STANDARD

---

- ▶ DBMS\_STANDARD, STANDARD – реализация всех базовых функций PL/SQL



# DBMS\_STATS

- ▶ DBMS\_STATS – сбор статистики
- ▶ статистические данные используются оптимизатором CBO (Cost Based Optimizer)

```
exec dbms_stats.gather_table_stats(User, 'STUDENT_GRADES', cascade => TRUE);
select * from DBA_TAB_STATS_HISTORY where table_name = 'STUDENT_GRADES';
|
```

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0,047 seconds

	OWNER	TABLE_NAME	PARTITION_NAME	SUBPARTITION_NAME	STATS_UPDATE_TIME
1	SYSTEM	STUDENT_GRADES	(null)	(null)	17.11.17 22:00:18,31
2	SYSTEM	STUDENT_GRADES	(null)	(null)	11.12.17 02:34:21,51

# DBMS\_TRANSACTION

---

- ▶ DBMS\_TRANSACTION – пакет управления транзакциями
- ▶ READ ONLY
- ▶ READ WRITE
- ▶ \*SET ISOLATION LEVEL READ COMMITTED
- ▶ \*SET ISOLATION LEVEL SERIALIZABLE
- ▶ \*LOCK TABLE

```
----DBMS_TRANSACTION
BEGIN
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
    INSERT INTO ST VALUES ('FIRST', 'STUDENT');
    DBMS_TRANSACTION.COMMIT;
END;
```



# Блокировки

---

- ▶ Блокировка – механизм, используемый для управления одновременным доступом к общему ресурсу
- ▶ ORACLE блокирует данные на уровне строк
- ▶ Используются блокировки и других уровней



# Блокировки

---

- ▶ Блокировка – механизм, используемый для управления одновременным доступом к общему ресурсу
- ▶ ORACLE блокирует данные таблицы на уровне строк
- ▶ Используются блокировки и других уровней
- ▶ Цель блокировок - обеспечить целостность и согласованность данных
- ▶ Пессимистическое и оптимистическое блокирование
- ▶ Уровни блокирования – монопольный и разделяемый
- ▶ Взаимные блокировки (deadlocks)
- ▶ Эскалация блокировок – преобразование блокировок



## DBMS\_TYPES

---

- ▶ DBMS\_TYPES – константы, которые представляют собой встроенные и определяемые пользователем типы



# DBMS.Utility

- ▶ DBMS.Utility – набор полезных утилит

```
DECLARE
    X VARCHAR2(20);
    Y VARCHAR2(200);
BEGIN
    X := 'SECRET TEXT';
    Y := DBMS_UTLILITY.GET_HASH_VALUE (X, 17, 1073741824);
    dbms_output.put_line(Y);
END;
```

## DBMS\_WARNING

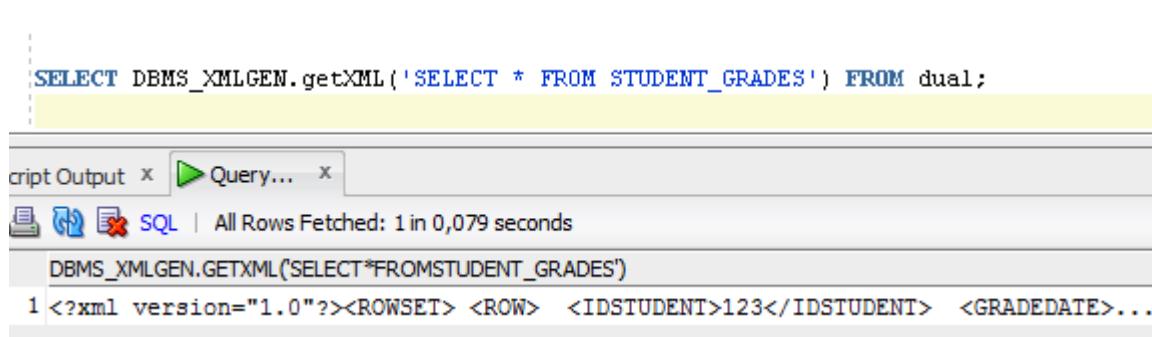
---

- ▶ DBMS\_WARNING – пакет для установки/сброса уровней предупреждений компилятора PL/SQL



# DBMS\_XMLGEN, DBMS\_XMLQUERY

- ▶ DBMS\_XMLGEN, DBMS\_XMLQUERY – пакеты для работы с XML – извлечение данных в формате XML из таблиц и процедур



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there is a code editor window containing the following SQL query:

```
SELECT DBMS_XMLGEN.getXML('SELECT * FROM STUDENT_GRADES') FROM dual;
```

Below the code editor is the SQL tab of the results panel, which displays the output of the query. The output shows the XML representation of the data from the STUDENT\_GRADES table:

```
DBMS_XMLGEN.GETXML('SELECT*FROMSTUDENT_GRADES')
1 <?xml version="1.0"?><ROWSET> <ROW> <IDSTUDENT>123</IDSTUDENT> <GRADEDATE>...
```

# DBMS\_XPLAN

- ▶ DBMS\_XPLAN – получение плана выполнения оператора

```
EXPLAIN PLAN FOR select * from STUDENT_GRADES WHERE IDSTUDENT = 123;
SELECT * FROM TABLE(dbms_xplan.display);
```

The screenshot shows the Oracle SQL Developer interface with the 'Query Result' tab selected. The results display the execution plan for the given SQL statement.

```
Script Output x Query Result x
SQL | All Rows Fetched: 13 in 0,108 seconds
PLAN_TABLE_OUTPUT
1 Plan hash value: 2030672728
2
3 -----
4 | Id  | Operation          | Name           | Rows  | Bytes | Cost (%CPU) | Time      |
5 -----
6 |   0 | SELECT STATEMENT   |                | 3    | 45   | 2 (0) | 00:00:01 |
7 |* 1 |  TABLE ACCESS FULL | STUDENT_GRADES | 3    | 45   | 2 (0) | 00:00:01 |
8 -----
9
10 Predicate Information (identified by operation id):
11 -----
12
13 1 - filter(TO_NUMBER("IDSTUDENT")=123)
```

## UTL\_I18N, UTL\_LMS

---

- ▶ Globalization Development Kit – пакеты UTL\_I18N и UTL\_LMS для разработки многоязычных приложений:
  - ▶ UTL\_I18N – территориальные настройки пользователя
  - ▶ UTL\_LMS – пакет обработки ошибок



# Вопросы проектирования многоязычных приложений

---

- ▶ Размерность переменных
- ▶ Порядок сортировки
- ▶ Формат денежных величин
- ▶ Формат даты и времени
- ▶ Поисковые системы



# Терминология

---

- ▶ Глобализация – стратегия разработки приложения, направленная на независимость от локального контекста:
  - ▶ Интернационализация – проектирование приложений, направленных на работу во многих локальных контекстах
  - ▶ Локализация – обеспечение работы приложения в каждом конкретном локальном контексте



# Терминология

---

- ▶ Кодовая точка – уникальный идентификатор, определяется Unicode, который представляет символ или может объединяться для представления составных символов
- ▶ Кодировка – отображение между символами и кодовыми точками
- ▶ Набор символов – множество символов, относящихся к некоторому региону
- ▶ Глиф – графическое представление кодовой точки
- ▶ NLS – National Language Support – Globalization Support



# Наборы символов

---

- ▶ Основной (database character set)
- ▶ Дополнительный (national character set)
  - ▶ UTF-8 – размер от 1 до 3 байт
  - ▶ UTF-16 – размер 2 байта
  - ▶ Дополнительные символы – 4 байта
- ▶ Устанавливаются при создании БД
- ▶ Изменяются alter database (national) character set



# Наборы символов

---

- ▶ Основной набор символов используется для:
  - ▶ хранения символьных типов `char`, `varchar2`, `clob` и `long`
  - ▶ описания имен объектов, переменных
  - ▶ Ввода и хранения PL/SQL модулей
- ▶ Дополнительный набор символов используется для:
  - ▶ хранения символьных типов `nchar`, `nvarchar2`, `nclob`
- ▶ Кроме символов алфавита в набор включаются знаки препинания, числа, символы денежных единиц и пр.



# Определение наборов символов

```
SELECT * FROM NLS_INSTANCE_PARAMETERS;
```

PARAMETER	VALUE
1 NLS_LANGUAGE	AMERICAN
2 NLS_TERRITORY	AMERICA
3 NLS_SORT	(null)
4 NLS_DATE_LANGUAGE	(null)
5 NLS_DATE_FORMAT	(null)
6 NLS_CURRENCY	(null)
7 NLS_NUMERIC_CHARACTERS	(null)
8 NLS_ISO_CURRENCY	(null)
9 NLS_CALENDAR	(null)
10 NLS_TIME_FORMAT	(null)
11 NLS_TIMESTAMP_FORMAT	(null)
12 NLS_TIME_TZ_FORMAT	(null)
13 NLS_TIMESTAMP_TZ_FORMAT	(null)
14 NLS_DUAL_CURRENCY	(null)
15 NLS_COMP	BINARY
16 NLS_LENGTH_SEMANTICS	BYTE
17 NLS_NCHAR_CONV_EXCP	FALSE

```
SELECT * FROM NLS_DATABASE_PARAMETERS;
```

PARAMETER	VALUE
1 NLS_RDBMS_VERSION	12.1.0.1.0
2 NLS_NCHAR_CONV_EXCP	FALSE
3 NLS_LENGTH_SEMANTICS	BYTE
4 NLS_COMP	BINARY
5 NLS_DUAL_CURRENCY	\$
6 NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
7 NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
8 NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
9 NLS_TIME_FORMAT	HH.MI.SSXFF AM
10 NLS_SORT	BINARY
11 NLS_DATE_LANGUAGE	AMERICAN
12 NLS_DATE_FORMAT	DD-MON-RR
13 NLS_CALENDAR	GREGORIAN
14 NLS_NUMERIC_CHARACTERS	.,
15 NLS_NCHAR_CHARACTERSET	AL16UTF16
16 NLS_CHARACTERSET	AL32UTF8
17 NLS_ISO_CURRENCY	AMERICA
18 NLS_CURRENCY	\$
19 NLS_TERRITORY	AMERICA
20 NLS_LANGUAGE	AMERICAN

# Определение наборов символов

```
SELECT * FROM NLS_SESSION_PARAMETERS;
```

Script Output x Query Result x

SQL | All Rows Fetched: 17 in 0,016 seconds

PARAMETER	VALUE
1 NLS_LANGUAGE	RUSSIAN
2 NLS_TERRITORY	RUSSIA
3 NLS_CURRENCY	p.
4 NLS_ISO_CURRENCY	RUSSIA
5 NLS_NUMERIC_CHARACTERS	,
6 NLS_CALENDAR	GREGORIAN
7 NLS_DATE_FORMAT	DD.MM.RR
8 NLS_DATE_LANGUAGE	RUSSIAN
9 NLS_SORT	RUSSIAN
10 NLS_TIME_FORMAT	HH24:MI:SSXFF
11 NLS_TIMESTAMP_FORMAT	DD.MM.RR HH24:MI:SSXFF
12 NLS_TIME_TZ_FORMAT	HH24:MI:SSXFF TZR
13 NLS_TIMESTAMP_TZ_FORMAT	DD.MM.RR HH24:MI:SSXFF TZR
14 NLS_DUAL_CURRENCY	p.
15 NLS_COMP	BINARY
16 NLS_LENGTH_SEMANTICS	BYTE
17 NLS_NCHAR_CONV_EXCP	FALSE

# Определение наборов символов

MSHELP_TOOLS	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1\MSHELP_TOOLS
NLS_LANG	REG_SZ	AMERICAN_AMERICA.WE8MSWIN1252
OLEDB	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1\OLEDB
OMTSRECO_PORT	REG_EXPAND_SZ	2030
ORA_ORCL_AUTOSTART	REG_EXPAND_SZ	TRUE
ORA_ORCL_SHUTDOWN	REG_EXPAND_SZ	TRUE
ORA_ORCL_SHUTDOWN_TIMEOUT	REG_EXPAND_SZ	90
ORA_ORCL_SHUTDOWNTYPE	REG_EXPAND_SZ	immediate
ORACLE_BASE	REG_SZ	C:\app\Oracle
ORACLE_BUNDLE_NAME	REG_SZ	Enterprise
ORACLE_GROUP_NAME	REG_SZ	Oracle - OraDB12Home1
ORACLE_HOME	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1
ORACLE_HOME_KEY	REG_SZ	SOFTWARE\ORACLE\KEY_OraDB12Home1
ORACLE_HOME_NAME	REG_SZ	OraDB12Home1
ORACLE_HOME_TYPE	REG_SZ	1
ORACLE_SID	REG_SZ	orcl
ORACLE_SVCUSER	REG_SZ	NT AUTHORITY\SYSTEMA
ORACLE_SVCUSER_PWDREQ	REG_SZ	0
RDBMS_ARCHIVE	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1\RDBMS_ARCHIVE
RDBMS_CONTROL	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1\RDBMS_CONTROL
SQLPATH	REG_SZ	C:\app\Oracle\product\12.1.0\dbhome_1\SQLPATH

# Определение наборов символов

---

- ▶ Переменная окружения **NLS\_LANG**:
- ▶ **NLS\_LANG = language\_territory.charset**
  - ▶ Язык (**LANGUAGE**) – имена месяцев, имена дней, направление текста, сокращения для времени и дат. По умолчанию **AMERICAN**
  - ▶ Территория (**TERRITORY**) – настройки календаря, формат даты, формат денежной единицы. Если не указан, то будет взято значение, соответствующее языку (для **RUSSIAN - CIS**)
  - ▶ Набор символов (**CHARACTER SET**) – отображение символов, отображение и конвертация заглавных букв, порядок замещения символов при преобразовании. Каждому языку поставлен в соответствие набор символов по умолчанию



# Порядок сортировки строк

---

- ▶ Двоичная
- ▶ Одноязычная
- ▶ Многоязычная



# Порядок сортировки строк

```
DECLARE
    retval utl_i18n.string_array;
    cnt   PLS_INTEGER;
BEGIN
    retval := utl_i18n.get_local_time_zones('AMERICA');
    dbms_output.put('Count = ');
    dbms_output.put_line(retval.LAST-retval.FIRST+1);
    cnt := retval.FIRST;
    WHILE cnt IS NOT NULL LOOP
        dbms_output.put_line(retval(cnt));
        cnt := retval.NEXT(cnt);
    END LOOP;
END;
```

```
Count = 8
America/New_York
America/Indianapolis
America/Chicago
America/Denver
America/Phoenix
America/Los_Angeles
America/Anchorage
Pacific/Honolulu
```

```
DECLARE
    retval utl_i18n.string_array;
    cnt   PLS_INTEGER;
BEGIN
    retval := utl_i18n.get_local_time_zones('RUSSIA');
    dbms_output.put('Count = ');
    dbms_output.put_line(retval.LAST-retval.FIRST+1);
    cnt := retval.FIRST;
    WHILE cnt IS NOT NULL LOOP
        dbms_output.put_line(retval(cnt));
        cnt := retval.NEXT(cnt);
    END LOOP;
END;
```

```
Count = 9
Europe/Moscow
Asia/Yekaterinburg
Asia/Novosibirsk
Asia/Krasnoyarsk
Asia/Irkutsk
Asia/Yakutsk
Asia/Vladivostok
Asia/Magadan
Asia/Kamchatka
```



# Порядок сортировки строк

```
DECLARE
    retval utl_i18n.string_array;
    cnt   PLS_INTEGER;
BEGIN
    retval := utl_i18n.get_local_languages('BELARUS');
    dbms_output.put('Count = ');
    dbms_output.put_line(retval.LAST-retval.FIRST+1);
    cnt := retval.FIRST;
    WHILE cnt IS NOT NULL LOOP
        dbms_output.put_line(retval(cnt));
        cnt := retval.NEXT(cnt);
    END LOOP;
END;

DECLARE
    retval utl_i18n.string_array;
    cnt   PLS_INTEGER;
BEGIN
    retval := utl_i18n.get_local_territories('FRENCH');
    dbms_output.put('Count = ');
    dbms_output.put_line(retval.LAST-retval.FIRST+1);
    cnt := retval.FIRST;
    WHILE cnt IS NOT NULL LOOP
        dbms_output.put_line(retval(cnt));
        cnt := retval.NEXT(cnt);
    END LOOP;
END;
```

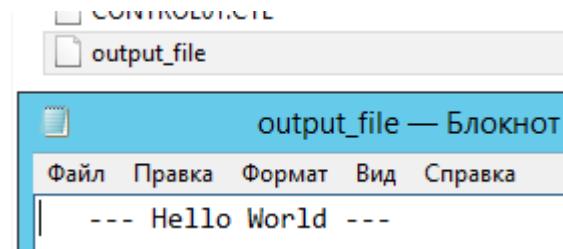
Count = 2  
BELARUSIAN  
RUSSIAN

Count = 13  
FRANCE  
BELGIUM  
CAMEROON  
CANADA  
CONGO BRAZZAVILLE  
CONGO KINSHASA  
DJIBOUTI  
GABON  
IVORY COAST  
LUXEMBOURG  
MAURITANIA  
SENEGAL  
SWITZERLAND

# UTL\_FILE

- ▶ UTL\_FILE – пакет, обеспечивающий файловый ввод/вывод

```
DECLARE
    file_Handle UTL_FILE.FILE_TYPE;
BEGIN
    file_Handle := UTL_FILE.FOPEN('DEST_DIR_1', 'output_file', 'W');
    UTL_FILE.PUTF(file_Handle, ' --- Hello World ---\n');
    UTL_FILE.FCLOSE(file_Handle);
EXCEPTION
    WHEN utl_file.invalid_path THEN
        raise_application_error(-20000, 'ERROR: Invalid path for file.');
END;
```



## UTL\_URL

---

- ▶ UTL\_URL – извлечение адреса из строки и наоборот



# UTL\_HTTP

---

- ▶ UTL\_HTTP – загрузка данных веб-страниц



# Вопросы?

---

