

Администрирование баз данных и приложений

**Архитектура ORACLE:
Внешняя память**

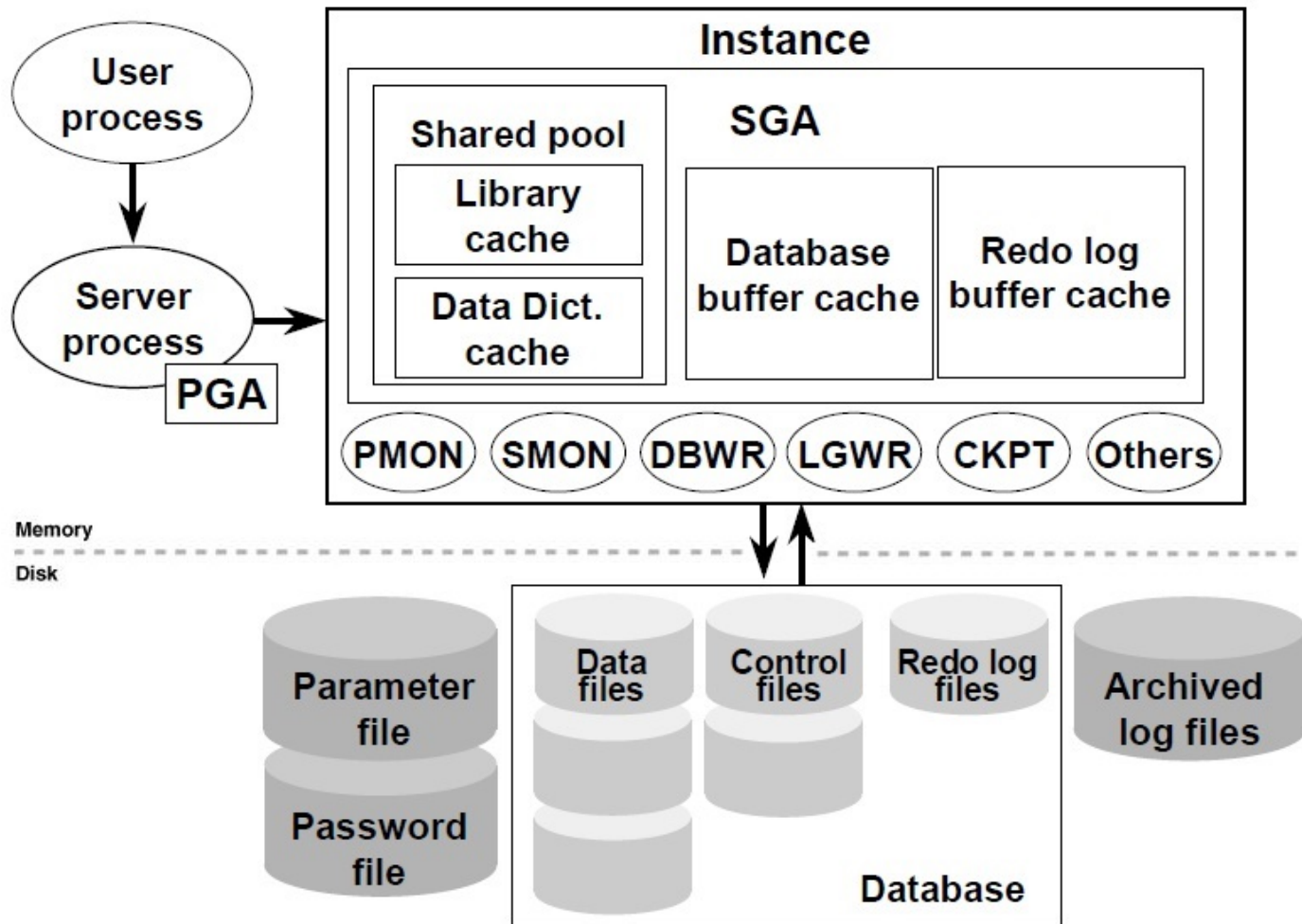
Лекция 3

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

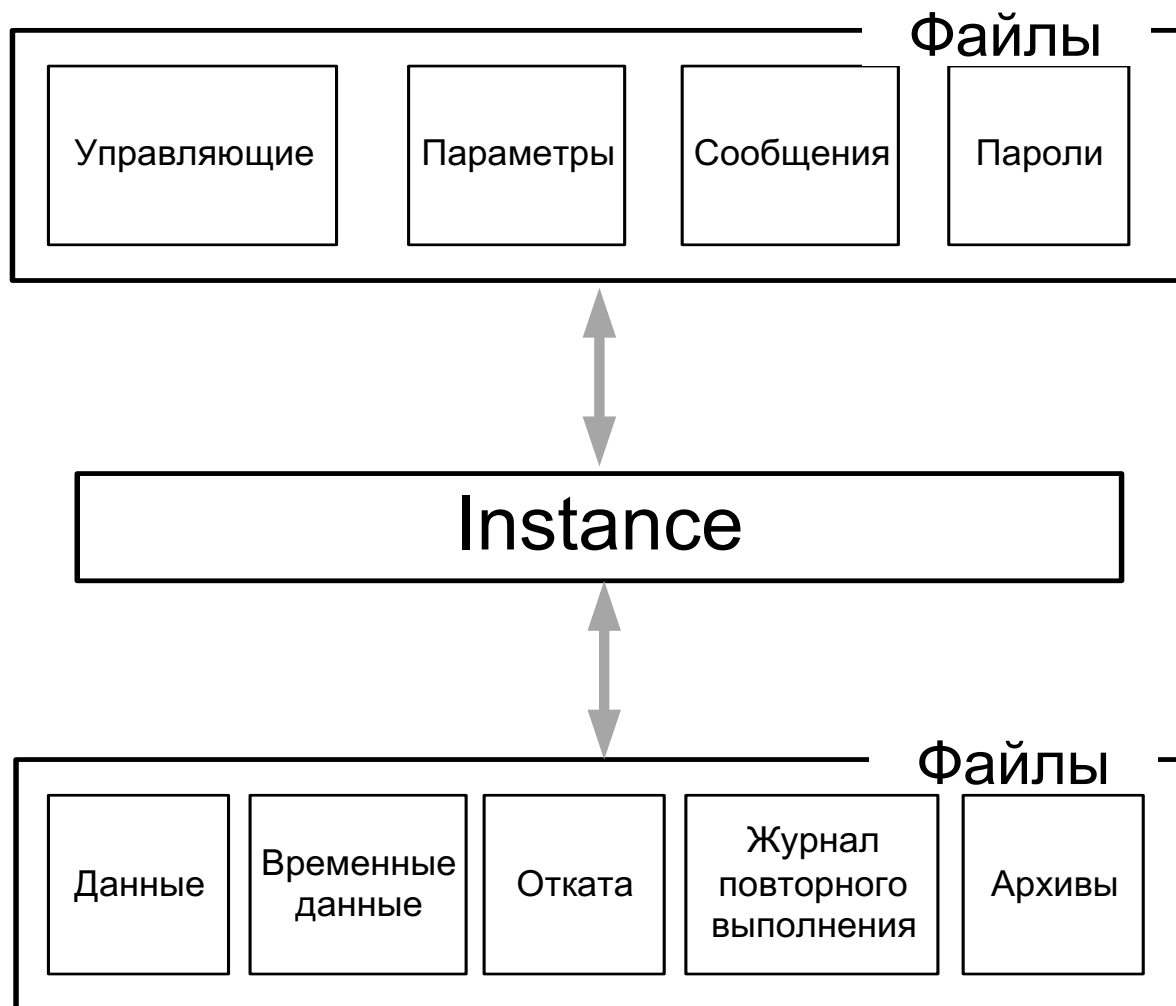
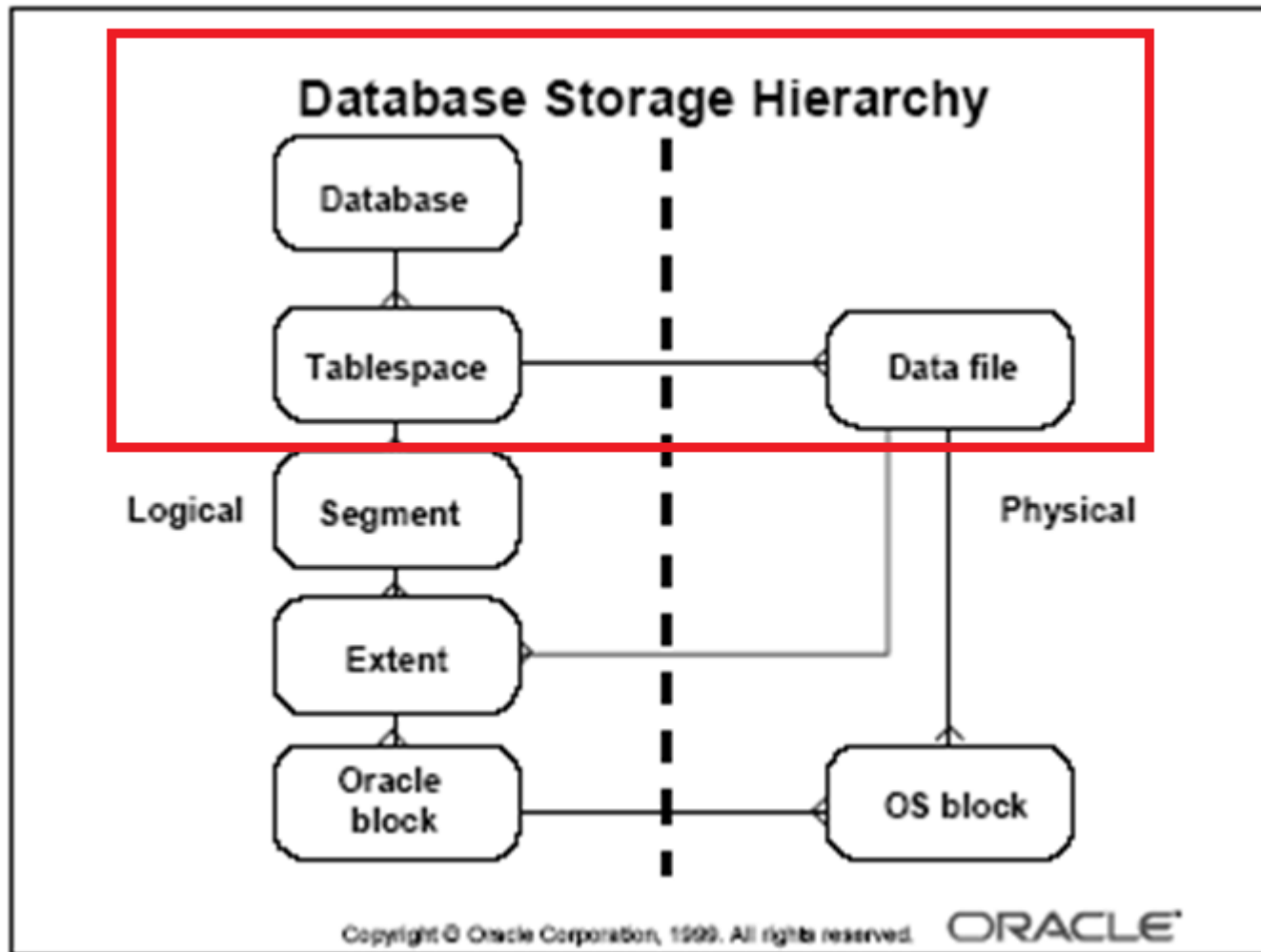


Схема архитектуры внешней памяти



Месторасположение файлов базы данных

Oracle Database 12c Release 1 Installer - Installing database - Step 8 of 12

Typical Install Configuration

Perform full database installation with basic configuration.

Oracle base: C:\app\ora_install_user Browse...

Software location: C:\app\ora_install_user\product\12.1.0\dbhome_1 Browse...

Storage type: File system

Database file location: C:\app\ora_install_user\oradata Browse...

Database edition: Enterprise Edition (6.0GB)

Global database name: orcl.be.by

Administrative password:

Confirm password: ?

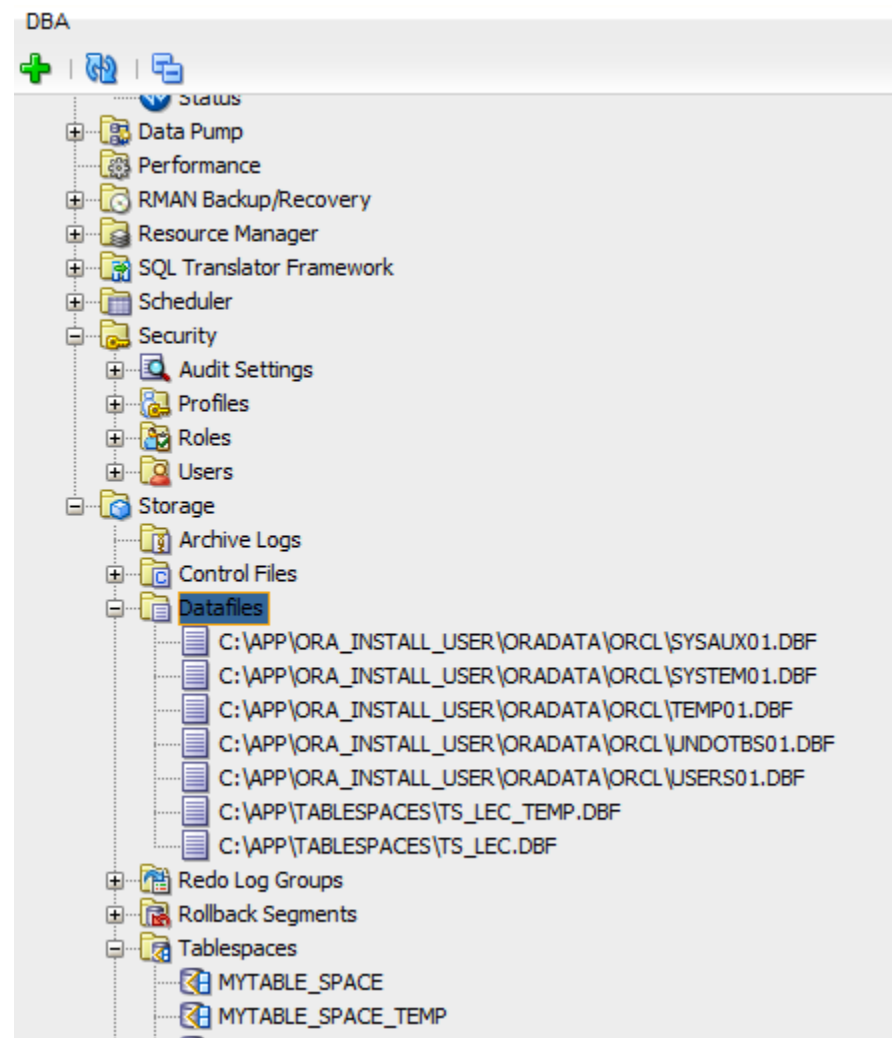
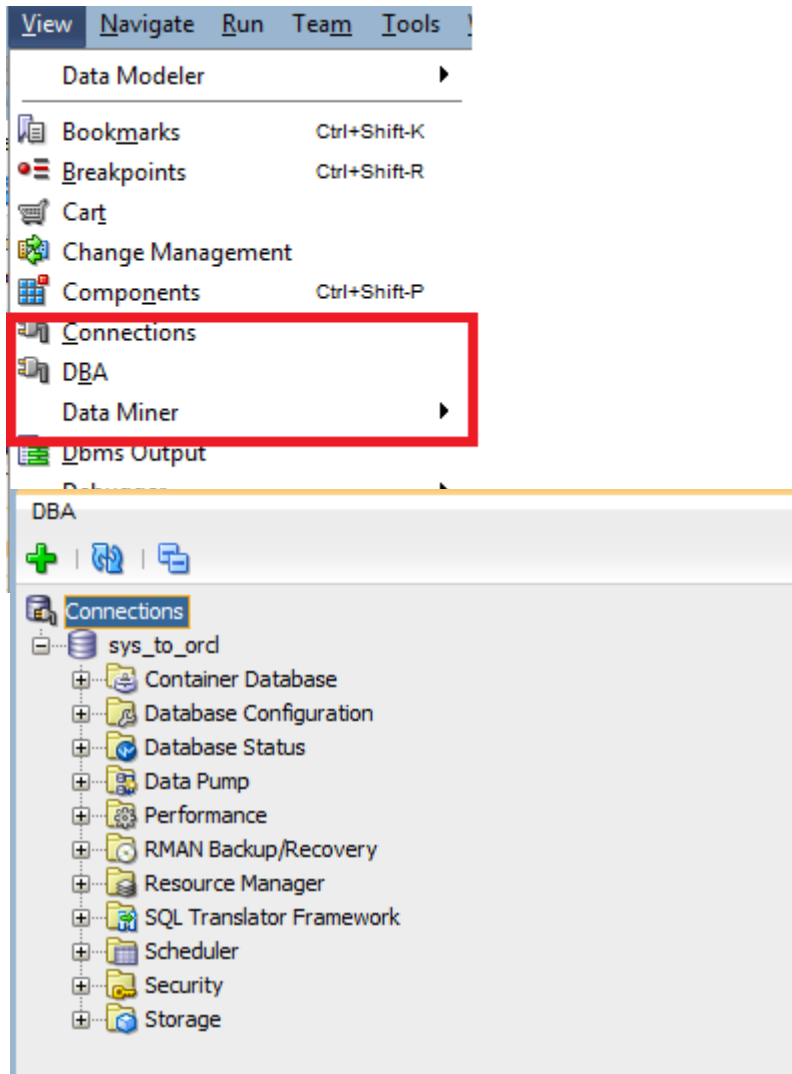
Service name:

☒ Create as Container database

Pluggable database name: pdborcl

Help < Back Next > Install Cancel

Месторасположение файлов базы данных

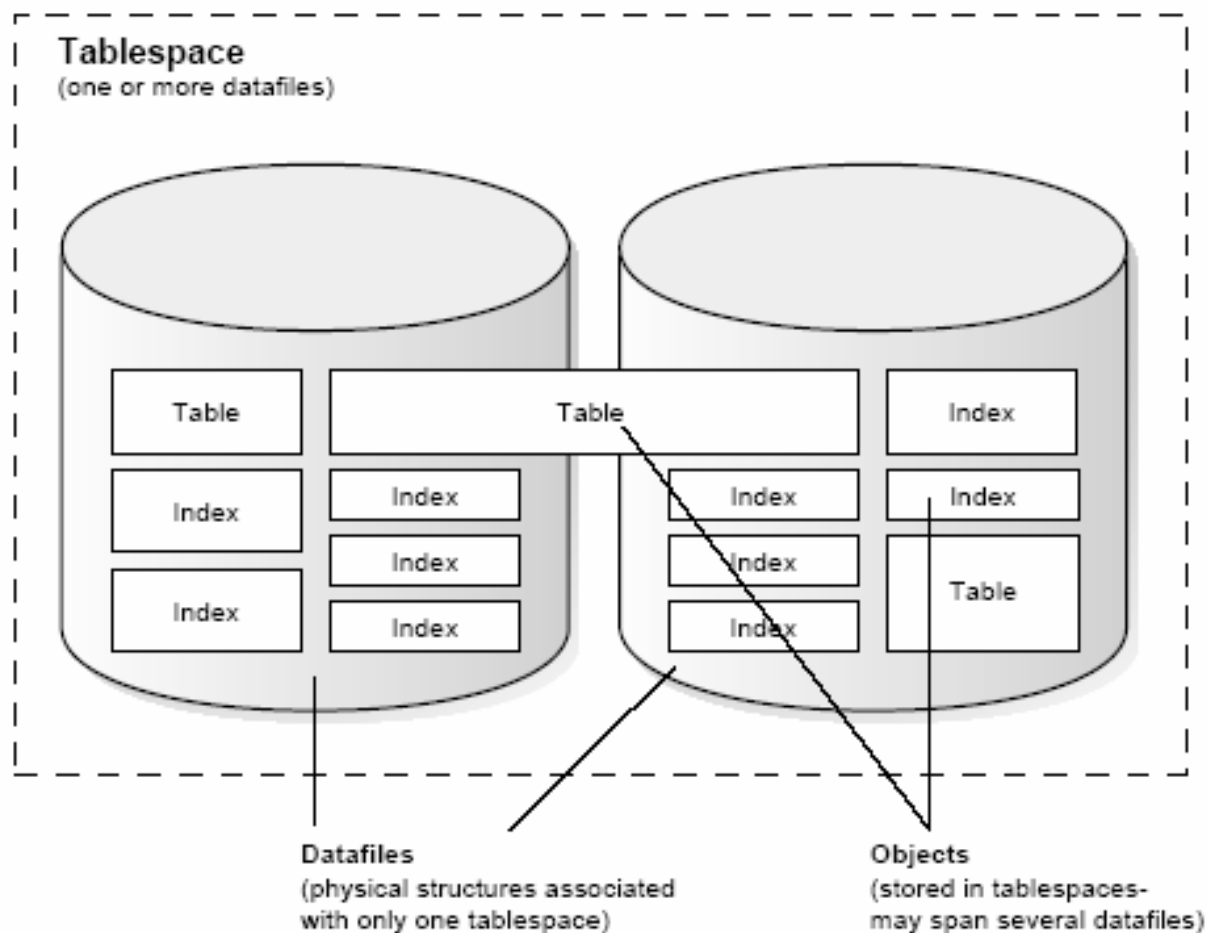


Табличное пространство

- ▶ 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

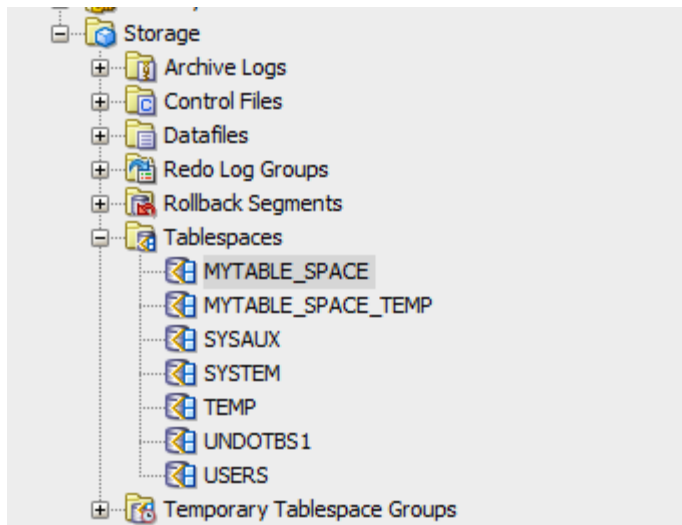


Предопределенные табличные пространства

- ▶ **SYSTEM** используется для управления БД, содержит словарь базы данных, стандартные пакеты процедур.
- ▶ **SYS_AUX** – 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

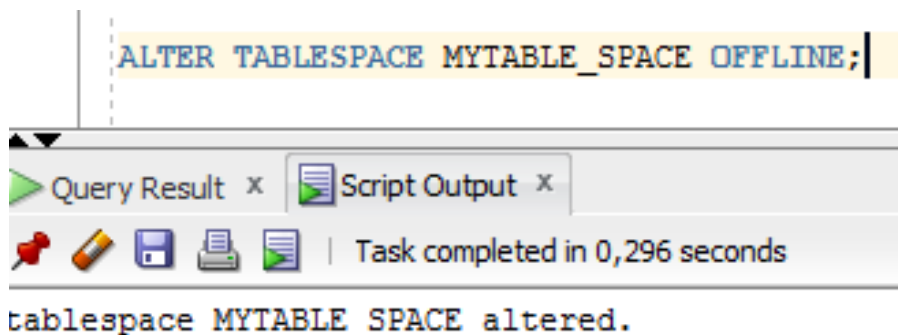
```
CREATE SMALLFILE TABLESPACE TBLSPC_1
DATAFILE
'C:\app\Tablespaces\DTBF_1.DBF' SIZE 10M REUSE AUTOEXTEND ON NEXT 2M MAXSIZE 100M ,
'C:\app\Tablespaces\DTBF_1.DBF' SIZE 10M REUSE AUTOEXTEND ON NEXT 2M MAXSIZE 100M
BLOCKSIZE 8192
LOGGING
ONLINE
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;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

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

Query Result x				
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



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 установке 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_file_dest='C:\app\oracle\admin\1305\adump'  
*.audit_trail='db'  
*.compatible='12.1.0.0.0'  
*.control_files='C:\app\oracle\oradata\ISUS\control01.ctl', 'C:\app\  
*.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

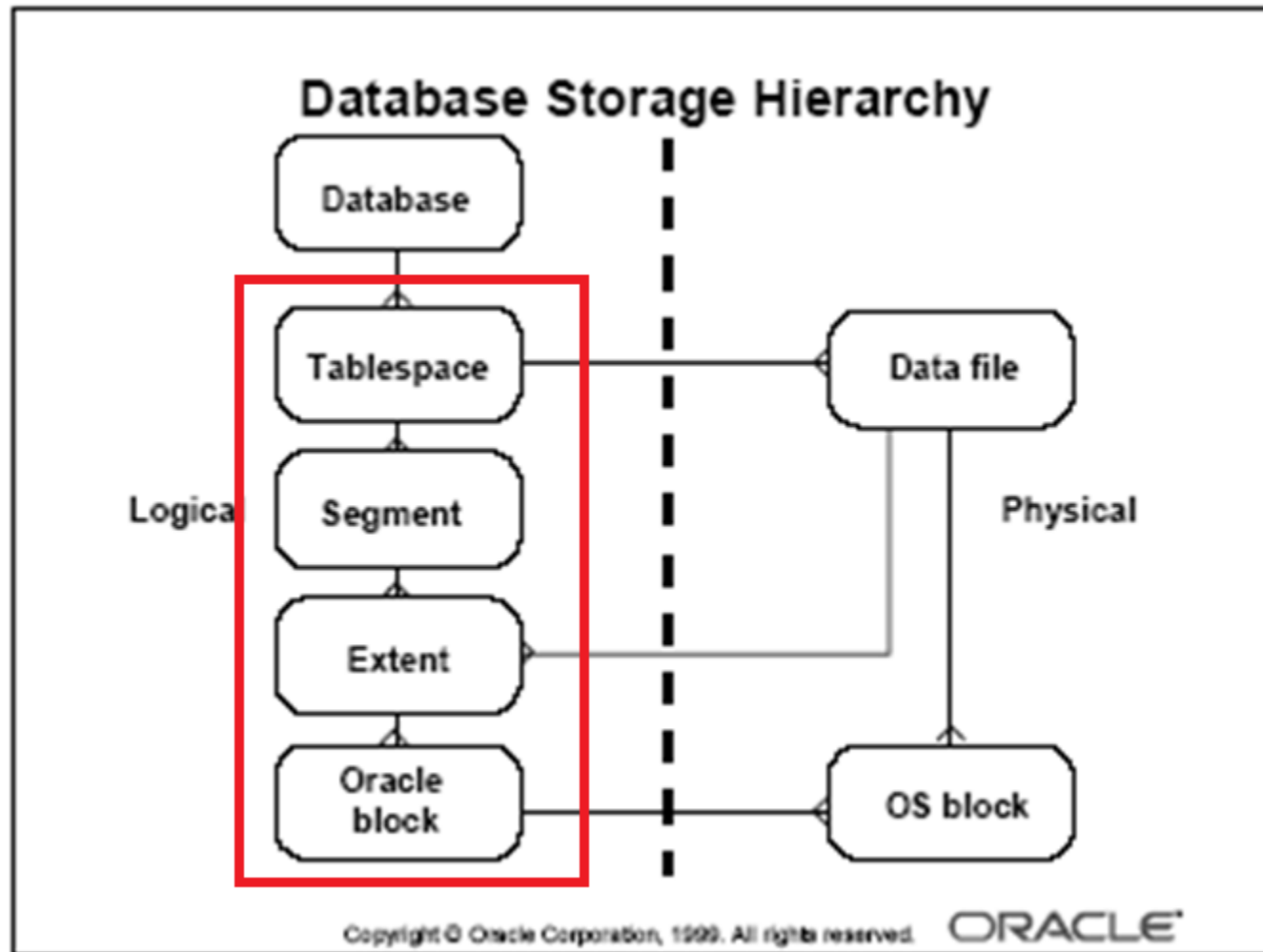
8 rows selected.

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.



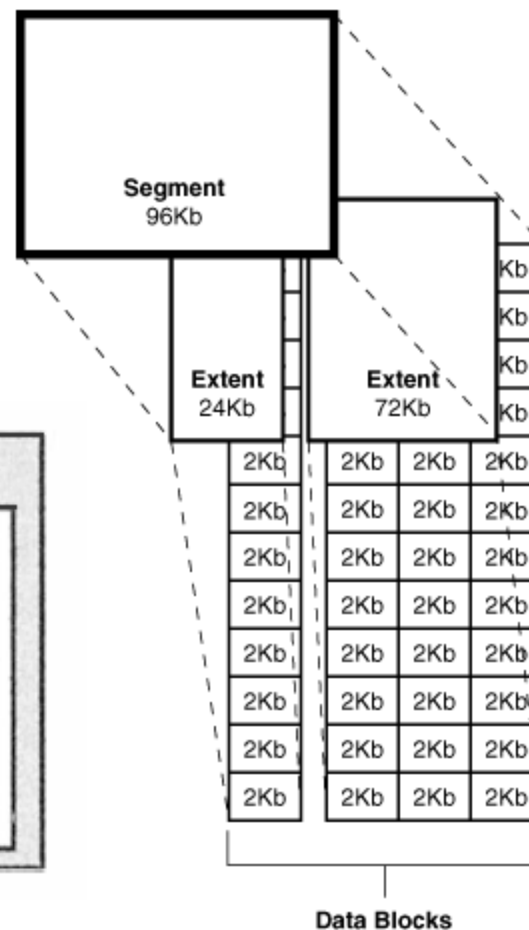
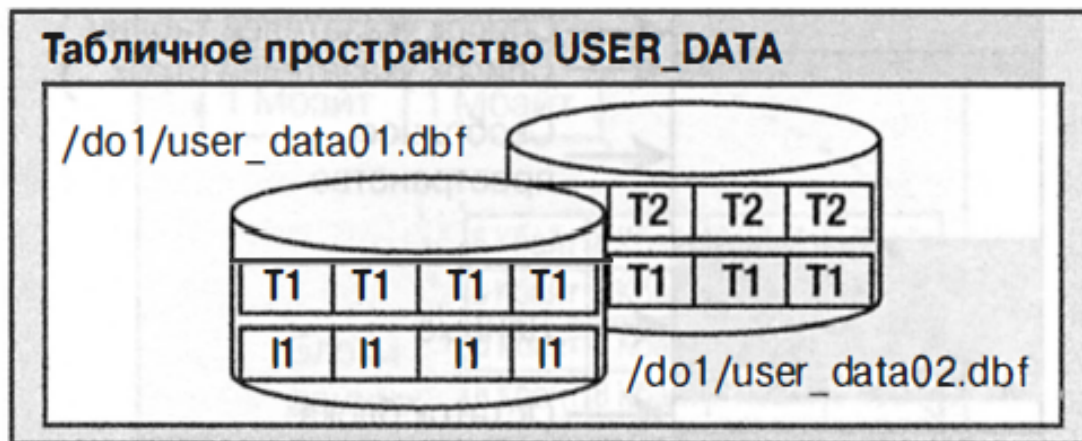
Блоки

[illegible]

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 экстенда
       initial_extent/block_size, -- кол. блоков в экстенде
       extent_management,   -- управление экстендами
       segment_space_management, -- управление сегментами
       bigfile
```

```
from dba_tablespaces;
```

	<small>TABLESPACE_NAME</small>	<small>BLOCK_SIZE</small>	<small>INITIAL_EXTENT</small>	<small>INITIAL_EXTENT/BLOCK_SIZE</small>	<small>EXTENT_MANAGEMENT</small>	<small>SEGMENT_SPACE_MANAGEMENT</small>	<small>BIGFILE</small>
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;
```

A
Z

сегменты есть в SVV_TBS01?

1 нет

Сегмент

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

```
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

<u>deferred segment creation</u>	boolean	<u>TRUE</u>
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
1	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
1	ISUS_B	XXX01	TABLE	SVV_TBS01	65536	8	DEFAULT

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

	TABLE_NAME	TABLESPACE_NAME
1	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 64K).
- ▶ 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 экстенды одного размера
  logging                                             -- журналирование/nologging (ум)
  online                                             -- offline/online(ум)
  extent management local                           -- локальное управление экстендами
  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';
```

R2	SEGMENT_NAME	R2	PARTITION_NAME	R2	SEGMENT_TYPE	R2	TABLESPACE_NAME	R2	EXTENT_ID	R2	BYTES	R2	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?

