|  |
| --- |
| **Лабораторная работа № 1. Администрирование СУБД Oracle. Управление доступом.** |
| **Oracle Database 21c Express Edition - установка Express Edition**  **( для загрузки free -** предварительно необходимо зарегистрироваться)  [**https://www.oracle.com/database/technologies/xe-downloads.html**](https://www.oracle.com/database/technologies/xe-downloads.html)  Существует много инструментов для работы с Oracle. Два стандартных инструмента это SQL \*Plus и SQL Developer. Они предоставляются компанией Oracle и подходят для администрирования и разработки. У SQL Developer больше функционал, но он требует графической подсистемы, а SQL \*Plus можно использовать в режиме командной строки.  Oracle Database представляет собой реляционную базу данных, которая хранит и извлекает коллекции связанной информации. В реляционной базе данных, объекты хранения информации организуются в структуры, называемые таблицами. Каждая таблица содержит строки (записи), которые состоят из столбцов (полей). Таблицы хранятся в базе данных в структурах, называемых схемами. Схемы - это логической структуры данных, в которых пользователи базы данных хранят свои таблицы и другие объекты.  Существует два способа подключения к Oracle в качестве администратора с использованием SQL\*Plus   1. connect sys as sysdba 2. connect system/manager(пароль)   **SYS AND SYSTEM**  учетные записи администраторов создаются автоматически при установке Oracle Database. Оба они созданы с паролем, который вы указали при установке, и им автоматически предоставляется роль DBA.   * SYS   Эта учетная запись может выполнять все административные функции. Все базовые (базовые) таблицы и представления для словаря данных базы данных хранятся в схеме SYS. Эти базовые таблицы и представления имеют решающее значение для работы базы данных Oracle. Для обеспечения целостности словаря данных таблицы в схеме SYS управляются только базой данных. Они никогда не должны быть изменены любым пользователем или администратором базы данных. Вы не должны создавать таблицы в схеме SYS.  Пользователю SYS предоставляется привилегия SYSDBA, которая позволяет пользователю выполнять высокоуровневые административные задачи, такие как резервное копирование и восстановление.   * SYSTEM   Эта учетная запись может выполнять все административные функции, кроме следующих:   * + Резервное копирование и восстановление   + Обновление базы данных   Хотя эту учетную запись можно использовать для выполнения повседневных административных задач, Oracle настоятельно рекомендует создать учетную запись именованных пользователей для администрирования базы данных Oracle, чтобы включить мониторинг активности базы данных.  **Системные привилегии SYSDBA и SYSOPER**  SYSDBA и SYSOPER - это административные привилегии, необходимые для выполнения административных операций высокого уровня, таких как создание, запуск, завершение работы, резервное копирование или восстановление базы данных. Системная привилегия SYSDBA предназначена для полностью уполномоченных администраторов баз данных, а системная привилегия SYSOPER позволяет пользователю выполнять основные операционные задачи, но без возможности просматривать пользовательские данные.  Системные привилегии SYSDBA и SYSOPER разрешают доступ к экземпляру базы данных, даже если база данных не открыта. Таким образом, контроль над этими привилегиями полностью находится за пределами самой базы данных. Этот элемент управления позволяет администратору, которому предоставлена ​​одна из этих привилегий, подключаться к экземпляру базы данных для запуска базы данных.   * Схема - это коллекция объектов базы данных, включая логические структуры, такие как таблицы, представления, последовательности, хранимые процедуры, синонимы, индексы, кластеры и ссылки на базы данных. * Пользователь владеет схемой. * Пользователь и схема имеют одно и то же имя. * Команда CREATE USER создает пользователя. Он также автоматически создает схему для этого пользователя. * Для всех намерений и целей вы можете считать пользователя схемой, а схему - пользователем.   Кроме того, пользователь может получить доступ к объектам в схемах, отличных от их собственных, если у него есть разрешение на это.  Права в базе данных Oracle  **Права (права)**— это возможность **выполнения конкретного типа SQL-оператора** или **доступа к объекту базы данных**, принадлежащему другому пользователю. В базе данных Oracle необходимо явно предоставить пользователю права для выполнения любых действий, включая подключение к базе данных или выборку, изменение и обновление данных в любой таблице, кроме собственной.  Существуют два основных типа прав Oracle: системные права и объектные права. Для предоставления пользователям как системных, так и объектных прав служит оператор GRANT.  Системные права  Системные права позволяют пользователю выполнить конкретное действие в базе данных либо действие с любым объектом схемы конкретного типа. Хороший пример первого типа системных полномочий — права, которые позволяют подключаться к базе данных, носящие название прав CONNECT. Другими правами этого типа являются права CREATE TABLESPACE, CREATE USER, DROP USER и ALTER USER. Второй класс системных прав предоставляет пользователям право на выполнение операций, которые влияют на объекты в любой схеме. Примерами этого типа системных прав служат ANALYZE ANY TABLE, GRANT ANY PRIVILEGE, NSERT ANY TABLE, GRANT ANY PRIVILEGE, INSERT ANY TABLE, DELETE ANY TABLE и т.п. Как видите, системные права являются очень мощным средством и выдача их не тому пользователю может оказать разрушительное влияние на базу данных. Ниже перечислены некоторые наиболее часто используемые права базы данных Oracle:   * *ADVISOR* * *ALTER DATABASE* * *ALTER SYSTEM* * *AUDIT SYSTEM* * *CREATE DATABASE LINK* * *CREATE TABLE* * *CREATE ANY INDEX* * *CREATE SESSION* * *CREATE TABLESPACE* * *CREATE USER* * *DROP USER* * *INSERT ANY TABLE*   Объектные права Объектные права — это полномочия по отношению к различным типам объектов базы данных. Объектные права дают пользователю возможность выполнять действия с конкретной таблицей, представлением, материализованным представлением, последовательностью, процедурой, функций или пакетом. Следовательно, всем пользователям базы данных нужны объектные права, даже если они не нуждаются в системных полномочиях. Существует ряд обычных системных прав, которые применяются ко всем объектам базы данных, и набор прав, применяемых только к определенным объектам. Для выдачи объектных прав можно использовать следующие SQL-операторы:   * *ALTER* * *SELECT* * *DELETE* * *EXECUTE* * *INSERT* * *REFERENCES* * *INDEX*   **Задание №1.**  Настройка SQL\*Plus. Необходимо включить режим ECHO и вывести протокол лаб. работы в файл *<Фамилия студента>.txt*. Этот файл является отчетом о проделанной лаб. работе.  Пример:  Следующий пример включает режим ECHO и ведет файл протокола spool. txt  SQL> set echo on  SQL> spool z:\lab1\_spool.txt  **Задание №2.**  Установить соединение с [базой данных](http://pandia.ru/text/category/bazi_dannih/) под пользователем system/manager. Создать пользователя базы данных с именем *user<Номер студента>*, паролем *user<Номер студента>*, назначить ему табличное пространство SYSTEM и задать ему неограниченное место в этом табличном пространстве. Пароли в Oracle должны начинаться с буквы, а не с цифры!  Пример:  SQL> connect system/masterkey…  SQL> create user user1 identified by user1  default tablespace system  quota unlimited on system;   * **[пользователь] [Username]** - Имя пользователя (название схемы). * **[пароль] [Password]** - Пароль для учетной записи. * **DEFAULT TABLESPACE** - Табличное пространство в котором будут находиться создаваемые в данной схеме объекты. Эта настройка не дает пользователю права создавать объекты - здесь устанавливается только значение по умолчанию. * **TEMPORARY TABLESPACE** - Табличное пространство, в котором находятся временные сегменты, используемые в процессе сортировки транзакций. * **QUOTA** - Позволяет пользователю сохранять объекты в указанном табличном пространстве, занимая там место вплоть до определенного в квоте общего размера.   - для версии 18 (обратите внимание на префикс к имени пользователя) С##...  SQL> CREATE USER с##user1 IDENTIFIED BY user1  2 DEFAULT TABLESPACE USERS QUOTA 10M ON USERS;  - посмотреть каталог пользователей можно:  SQL> SELECT USERNAME FROM SYS.DBA\_USERS;  - посмотреть структуру таблицы SYS.DBA\_USERS; можно  SQL> DESC SYS.DBA\_USERS  - изменить табличное пространство  SQL> ALTER USER c##user1 DEFAULT TABLESPACE USERS QUOTA 15M ON USERS;  SQL> ALTER USER с##user1 TEMPORARY TABLESPACE TEMP QUOTA 1M ON USERS;   * Посмотреть информацию из таблицы SYS.DBA\_USERS   SQL> SELECT USERNAME, USER\_ID, PASSWORD, ACCOUNT\_STATUS, DEFAULT\_TABLESPACE,  2 TEMPORARY\_TABLESPACE, PROFILE  3 FROM DBA\_USERS  4 WHERE USERNAME = 'с##user1';  **Задание №3**.  Установить соединение с базой данных под пользователем system/masterkey….  Предоставить пользователю *user<Номер студента>*роль CONNECT (возможность устанавливать соединение с базой данных) и системную привилегию CREATE TABLE (возможность создавать таблицы в собственной схеме).  Пример:  SQL> connect system/masterkey…  SQL> grant connect to user1;  SQL> grant create any table to user1;  **Задание №4.**  Установить соединение с базой данных под пользователем *user<Номер студента>*, вывести на экран имя текущего пользователя (команда SQL: select username from user\_users). Затем создать таблицу tab\_user<Номер студента>, в которой первый столбец col1 типа NUMBER, второй столбец col2 типа VARCHAR2(20). Затем внести в таблицу некоторые данные (минимум две записи) и выполнить завершение транзакции (команда COMMIT).  Пример:  SQL> connect user1/user1  SQL> select username from user\_users;  SQL> create table tab\_user1(col1 NUMBER, col2 VARCHAR2(20));  SQL> insert into tab\_user1 values (1, 'USER1');  SQL> insert into tab\_user1 values (2, 'USER1');  SQL> commit;  **Задание №5.**  Предоставить пользователю *user<Номер студента + 1>* следующие привилегии доступа к таблице tab\_user<Номер студента>: SELECT, INSERT, DELETE.  Пример:  SQL> grant select on user1.tab\_user1 to user2;  SQL> grant insert on user1.tab\_user1 to user2;  SQL> grant delete on user1.tab\_user1 to user2;  **Системные права** **наиболее часто используемые:** CREATE SESSION – право подключения к БД ALTER DATABASE – право изменения БД CREATE TABLESPACE – право создавать табличное пространтсво ALTER TABLESPACE – право изменять табличное пространтсво DROP TABLESPACE – право удалять табличное пространтсво CREATE TABLE – право создавать, изменять, удалять таблицы в своей схеме INSERT ANYTABLE – право добавлять данные в таблиц, которые не принадлежат учетной записи UPDATE ANYTABLE – право изменять данные в таблиц, которые не принадлежат учетной записи DELETE ANYTABLE – право удалять данные в таблиц, которые не принадлежат учетной записи SELECT ANYTABLE – право выборки данных из таблиц, которые не принадлежат учетной записи  **Синтаксис назначения прав:** GRANT privilege [,privilege…] TO User\_Name;  **Задание №6.**  Установить соединение с базой данных под пользователем *user<Номер студента + 1>* и просмотреть содержимое таблицы tab\_user<Номер студента>.  Затем необходимо добавить и удалить одну запись этой таблицы и выполнить завершение транзакции (команда COMMIT).  Примечание: необходимо указать имя схемы при обращении к этой таблице. Т. е. если вы находитесь в схеме пользователя user2 и хотите обратиться к таблице tab\_user1, находящейся в схеме пользователя user1, то необходимо указывать полное имя таблицы: user1.tab\_user1  Пример:  connect user2/user2  SQL> select \* from user1.tab\_user1;  SQL> insert into user1.tab\_user1 values (3, 'USER2');  SQL> delete from user1.tab\_user1 where col1=3;  SQL> commit;  **Задание №7.**  Установить соединение с базой данных под пользователем *user<Номер студента>*. Отменить для пользователя *user<Номер студента + 1>* следующие привилегии доступа к таблице tab\_user<Номер студента>: SELECT, INSERT, DELETE.  Пример:  SQL> connect user1/user1  SQL> revoke select on user1.tab\_user1 from user2;  SQL> revoke insert on user1.tab\_user1 from user2;  SQL> revoke delete on user1.tab\_user1 from user2;  **Задание №8.**  Повторить пункт №6 и объяснить полученные ошибки.  **Задание №9.**  Установить соединение с базой данных под пользователем *user<Номер студента>*. Попытаться просмотреть имена всех пользователей, зарегистрированных в базе данных. Объяснить полученное сообщение об ошибке.  Примечание: Для просмотра словаря данных необходимо обладать ролью SELECT\_CATALOG\_ROLE или DBA.  Пример:  SQL> connect user1/user1  SQL> select username from sys. dba\_users;  **Задание №10.**  Установить соединение с базой данных под пользователем system/manager. Предоставить пользователю *user<Номер студента>*роль DBA. Затем установить соединение с базой данных под пользователем *user<Номер студента>*. Попытаться просмотреть имена всех пользователей, зарегистрированных в базе данных.  Пример:  SQL> connect system/masterkey…  SQL> grant dba to user1;  SQL> connect user1/user1  SQL> select username from sys. dba\_users;  **Задание №11.**  Установить соединение с базой данных под пользователем system/manager. Изменить пароль пользователя *user<Номер студента>* на пароль test. Создать профиль prof\_user<Номер студента>, который устанавливает максимальное число неудачных попыток входа – 2. Затем необходимо назначить пользователю *user<Номер студента>* этот профиль.  Пример:  SQL> connect system/masterkey…  SQL> alter user user1 identified by test;  SQL> CREATE PROFILE prof\_user1 LIMIT  FAILED\_LOGIN\_ATTEMPTS 2;  SQL> alter user user1 profile prof\_user1;  SQL> ALTER USER c##user1 QUOTA 50M ON USERS  Пользователь изменен.  Теперь c##user1 не выйдет за пределы 50 Мб. С помощью профилей можно ограничить количество ресурсов системы и БД доступных для пользователя, а так же управлять ограничениями налагаемыми паролями. Если пользователю не назначен профиль по умолчанию, то будет использовать профиль **DEFAULT**. Ограничим профиль и назначим его нашему пользователю:  SQL>CREATE PROFILE c##user1 LIMIT PASSWORD\_LIFE\_TIME 180;  SQL>ALTER USER c##user1 PROFILE c##user1;  **Задание №12.**  Выполнить две неудачные попытки входа в базу данных под пользователем *user<Номер студента>*, указывая при этом неверный пароль. Затем попытаться установить соединение с базой данных под пользователем *user<Номер студента>*, указав верный пароль. Объяснить полученное сообщение об ошибке.  Пример:  SQL> connect user1/bad\_password @lab  SQL> connect user1/bad\_password@lab  SQL> connect user1/test  **Задание №13.**  Установить соединение с базой данных под пользователем system/manager. Разблокировать учетную запись пользователя *user<Номер студента>.*Установить соединение с базой данных под пользователем *user<Номер студента>*.  Пример:  SQL> connect system/masterkey…  SQL> alter user user1 account unlock;  SQL> connect user1/test  **Задание №14.**  **Посмотреть информацию о имеющихся привилегиях**  --Список системных привилегий для ролей  SELECT \* FROM ROLE\_SYS\_PRIVS;  --Привилегии на таблицы для ролей  SELECT \* FROM ROLE\_TAB\_PRIVS;  --Роли, доступные текущему пользователю (!!! Подсоединитесь под разными пользователями SYS SYSTEM HR!!!)  DESC USER\_ROLE\_PRIVS;  SELECT \* FROM USER\_ROLE\_PRIVS;  --Объектные привилегии доступные текущему пользователю  SELECT \* FROM USER\_TAB\_PRIVS\_RECD;  --Системные роли, доступные определенному пользователю (gодсоединитесь под пользователями SYS или SYSTEM )  SELECT \* FROM DBA\_ROLE\_PRIVS WHERE GRANTEE = 'User\_Name';  **Задание №15**.  Установить соединение с базой данных под пользователем system/manager. Каскадно удалить пользователя *user<Номер студента>.* Удалить профиль prof\_user<Номер студента>.  Пример:  SQL> connect system/masterkey…  SQL> drop user user1 cascade;  SQL> drop profile prof\_user1;  **Задание №16.**  Завершить протокол лаб. работы (команда spool off) и показать результаты преподавателю.  Пример:  SQL> spool off  **Задание №17. Ознакомиться.**  Настройка среды SQL\*Plus для разработчика приложений Oracle  Для **профессионального разработчика** (программиста) очень важно уметь **эффективно работать в среде SQL\*Plus**. Поэтому важно уметь выполнять **грамотную настройку и конфигурирование SQL\*Plus**. Почти во всех примерах в учебных пособиях и книгах тем или иным образом используется пакет DBMS\_OUT PUT. Чтобы можно было работать с DBMS\_OUTPUT, потребуется выдать следующую команду SQL\*Pius:  SQL> set serveroutput on  Частый ввод этой команды довольно быстро становится утомительным. К счастью, SQL\*Plus позволяет создать файл login. sql — сценарий, который выполняется при каждом запуске SQL\*Plus. Более того, можно определить переменную среды SQLPATH, что позволит находить этот сценарий независимо от того, в каком каталоге он хранится.  define \_editor=vi  set serveroutput on Size unlimited  set trimspool on  set long 5000  set linesize 100  set pagesize 9999  Ниже приведена аннотированная версия этого сценария.  1. **define \_editor=vi**. Определяет текстовый редактор, который SQL\*Plus будет использовать по умолчанию. Можете указать любой предпочитаемый текстовый редактор (не текстовый процессор), такой как Notepad или emacs.  2. **set serveroutput on size unlimi ted**. По умолчанию включает пакет DBMS\_ OUTPUT (следовательно, вводить каждый раз команду set serveroutput on не придется). Также устанавливает стандартный размер буфера в максимально возможное значение.  3. **set trimspool on**. При буферизации текста строки будут усекаться с отбрасыванием пробелов, поэтому они не будут иметь фиксированную длину. Если параметр trimspool установлен в off (по умолчанию), ширина буферизованных строк будет равна значению параметра linesize.  4. **set long 5000**. Устанавливает стандартное количество байтов, отображаемых при выборе столбцов LONG и CLOB.  5. **set linesize 100**. Устанавливает ширину строк, отображаемых SQL\*Plus, в 100 символов.  6. **set pagesize 9999**. Устанавливает параметр pagesize, который управляет тем, насколько часто SQL\*Plus выводит заголовки, в большое значение (мы будем получать один набор заголовков на страницу).  **Oracle server это экземпляр БД подключенный к БД**  Экземпляр БД – это область разделяемой памяти и набор процессов  БД – это набор файлов на диске  Пользовательская сессия – это пользовательский процесс соединённый с серверным процессом  Структуры памяти экземпляра БД  Разделяемая (общая) память называется system global area (SGA)  Неразделяемая (частная) память сессий – это program global area (PGA)  SGA состоит из подсруктур, часть из которых обязательные (кэш буфера БД (database  buffer cache), буфер логов, разделяемая область (shared pool)) и которые необязательные (large pool, Java pool, Streams pool)  Структуры SGA могут динамически изменять размера и управляться автоматически, за исключением буфера логов.  Процессы экземпляра БД  Серверный процесс (для сессии) запускается, когда пользователь подключается  Фоновые процессы запускаются, когда стартует экземпляр БД и существуют до его остановки  Серверные процесс ы читают информацию из БД; фоновые процессы записывают изменения в БД  Некоторые фоновые процессы присутствуют во всех серверах БД (SMOM, PMON, DBWn, LGWR, CKPT и MMON); остальные будут или не будут запущены в зависимости от найстроек сервера  Структуры хранения БД  Существуют три необходимых типа файлов в БД: controlfile, online redo log файлы и файлы данных  Controlfile хранит ограничители целостности и указатели для работы со всей БД  Online redo log файлы содержат последние вектора изменений (change vectors)  Сами данные хранятся в файлах данных  Дополнительные файлы как файл параметров запуска (parameter file pfile and spfile), файлы паролей, архивные логи (archive redo logs) и log and trace файлы.  Логические структуры (segment-ы) абстрагированы от физической информации на диске (файлов данных) с помощью понятия табличное пространство (tablespace)  Табличное пространство может включать в себя много файлов данных  Сегменты (таблицы, индексы и т.д.) состоят из списка extent-ов, каждый из которых есть не что иное, как набор блоков Oracle, которые в свою очередь состоят из одного или более блоков операционной системы.  Сегменты могут храниться в нескольких различных файлах данных  Использование представлений DBA для управления пользователями, ролями и правами  Основные представления словаря данных, которые можно применять для управления пользователями, полномочиями и ролями в базе данных, перечислены в таблице ниже.   |  |  | | --- | --- | | **Представление словаря данных** | **Описание** | | *DBA\_USERS* | Предоставляет информацию о пользователях. | | *DBA\_ROLES* | Отображает все роли в базе данных. | | *DBA\_COL\_PRIVS* | Отображает полномочия, предоставленные на уровне  столбцов. | | *DBA\_ROLE\_PRIVS* | Отображает пользователей и их роли. | | *DBA\_SYS\_PRIVS* | Отображает пользователей, которым предоставлены  системные полномочия. | | *DBA\_TAB\_PRIVS* | Отображает пользователей и их полномочия в таблицах. | | *ROLE\_ROLE\_PRIVS* | Отображает роли, предоставленные ролям. | | *ROLE\_SYS\_PRIVS* | Отображает системные роли, предоставленные ролям. | | *ROLE\_TAB\_PRIVS* | Отображает табличные полномочия, предоставленные  ролям. | | *SESSION\_PRIVS* | Отображает полномочия, которые в данный момент  включены для текущего сеанса. | | *SESSION\_ROLES* | Отображает роли, которые в данный момент включены  для текущего сеанса. |    Настройки выполнения скриптов. Действуют на протяжении сессии в SQLplus.  SET SERVEROUTPUT ON/OFF SIZE 1000000 FORMAT WORD\_WRAPPED - отображать вывод dbms\_output (размер буфера, байт) (разделение по строкам для длинного вывода)  SET TERMOUT ON/OFF - включить/отключить вывод на экран терминала  SET ECHO ON/OFF - включить/отключить отображение команд по мере выполнения  SET VERIFY ON/OFF - включить/отключить отображение замененных переменных  например, так:  old 9: AND ui.table\_name = UPPER('&table\_name')  new 9: AND ui.table\_name = UPPER('project\_hours')  SET FEEDBACK ON/OFF - включить/отключить вывод результата выполнения команд  например, так: 6 rows selected. Commit complete.  SET ESCAPE ON/OFF - включить/отключить поиск символов для экранирования  например: SQL> SET ESCAPE ON  SQL> DEFINE friends = "Joe \& Matt"  SET ESCAPE / - задать другой символ экранирования (по умолчанию \)  SET CONCAT ON/OFF - включить/отключить конкатенацию, символ конкатенации по умолчанию - точка "."  например: SQL> DEFINE sql\_type = "PL/"  SQL> PROMPT &sql\_type.SQL  PL/SQL  SET CONCAT ! - задать другой символ для конкатенации  SET DEFINE ON/OFF - включить/отключить замену переменных (&var, &&var)  (Вопрос на экзамене) Табличные пространства и файлы данных. Работа с табличными пространствами. Метаданные.  В базе данных ORACLE все данные хранятся в табличном пространстве из логической структуры, конечно же, есть логические структуры, такие как сегменты, области, блоки под табличным пространством. Из физической структуры он помещается в файл данных. Табличное пространство может состоять из нескольких файлов данных.  Как показано на следующем рисунке, база данных состоит из одного или нескольких табличных пространств, которые логически состоят из одного или нескольких сегментов и физически состоят из одного или нескольких файлов os.  https://images.cnitblog.com/blog/73542/201311/11232345-38d4cc57cfbc4d0695d4bb42157068ea.jpg |

Несколько табличных пространств, созданных по умолчанию в системе:

* Default Permanent Tablespace
* Табличное пространство SYSAUX
* Составное табличное пространство Temp
* Undo

**Постоянное табличное пространство** Храните постоянные данные, такие как таблицы, индексы и т. д.

**Временное табличное пространство** Невозможно хранить постоянные объекты, используемые для сохранения временных данных, сгенерированных при сортировке и группировке базы данных.

**UNDO табличное пространство** Сохраните зеркальное отображение перед изменением данных.

Как проверить, какие табличные пространства находятся в базе данных? Как просмотреть файл данных, соответствующий табличному пространству?

Просмотр табличного пространства позволяет просматривать основную информацию через следующие системные представления

-Содержит описательную информацию для всех табличных пространств в базе данных

SELECT \* FROM DBA\_TABLESPACES

-Содержит информацию описания табличного пространства текущего пользователя

SELECT \* FROM USER\_TABLESPACES

-Содержит имя табличного пространства и информацию о номере, полученную из контрольного файла

SELECT \* FROM V$TABLESPACE;

Просмотр файлов данных

-Содержит информацию описания файла данных и табличного пространства, к которому он принадлежит

SELECT \* FROM DBA\_DATA\_FILES

-Содержит информацию описания временного файла данных и табличного пространства, к которому он принадлежит

SELECT \* FROM DBA\_TEMP\_FILES

- Содержит основную информацию о файле данных, полученном из контрольного файла, включая имя и номер табличного пространства, к которому он принадлежит

SELECT \* FROM V$DATAFILE

- Содержит основную информацию обо всех временных файлах данных

SELECT \* FROM V$TEMPFILE

Следующий запрос можно использовать для определения табличного пространства и других параметров:

select df.tablespace\_name "Tablespace",

totalusedspace "Used MB",

(df.totalspace - tu.totalusedspace) "Free MB",

df.totalspace "Total MB",

round(100 \* ( (df.totalspace - tu.totalusedspace)/ df.totalspace)) "Pct. Free"

from (select tablespace\_name,

round(sum(bytes) / 1048576) TotalSpace

from dba\_data\_files

group by tablespace\_name) df,

(select round(sum(bytes)/(1024\*1024)) totalusedspace,

tablespace\_name

from dba\_segments

group by tablespace\_name) tu

where df.tablespace\_name = tu.tablespace\_name

and df.totalspace <> 0;

select owner,

segment\_name,

partition\_name,

segment\_type,

bytes / 1024/1024 "MB"

from dba\_segments

where owner = ‘user1’;

#### В Oracle база данных состоит из одного или нескольких табличных

#### пространств (table spaces). Каждое такое пространство строится на

#### одном или нескольких файлах данных. В одно табличное пространство

#### стараются помещать объекты с одинаковым поведением. Например,

#### для словаря базы выделяется отдельное табличное пространство,

#### обычно называемое системным.

#### Пользовательские данные желательно помещать отдельно от словаря.

#### Это уменьшит вероятность сбоя. Для, индексов следует иметь свои

#### табличные пространства.

#### Можно отключать отдельные табличные пространства и делать их

#### доступными только по чтению. Для больших сортировок можно создавать

#### временные табличные пространства.

#### Администратор должен выбрать состав, размеры табличных пространств и

#### определить, могут ли они расширяться, и какими порциями им будет

#### предоставляться свободное пространство дисковой памяти.

#### Табличные пространства состоят из сегментов, содержащих хранимые

#### объекты базы, например, таблицы, индексы. Каждому такому объекту положено

#### иметь свой сегмент, куда нет доступа данным других объектов.

#### Некоторые виды сегментов:

#### Сегменты данных: Каждой таблице (кроме кластеризованных и индексно-организованных) выделяется свой сегмент данных. Один или более сегментов может быть у внешних таблиц, глобальных временных таблиц и сегментированных таблиц. Для разделенной таблицы у каждого раздела есть свой сегмент данных. У каждого кластера есть сегмент данных общий для входящих в кластер таблиц..

#### Индексные сегменты: У каждого индекса есть индексный сегмент, который хранит все его данные. Для разделенного индекса у каждого раздела есть индексный сегмент.

#### Сегменты отката: Одно табличное пространство отката создается для каждого экземпляра базы данных. Эта табличное пространство содержит сегменты временно хранящие информацию отката, которая используется, в частности, для отката незафиксированных транзакций.

#### Врѐменные сегменты: Временные сегменты создаются когда инструкция SQL нуждается в дополнительном пространстве. Типичный пример – большая сортировка.

#### Сегменты состоят из экстентов, представляющих логически непрерывные или расположенные на диске непрерывно наборы блоков данных. Это ускоряет операции с блоками данных, входящими в состав экстента. Можно при работе с любым элементом данных, читать весь экстент, в надежде, что его данные скоро понадобятся. Размер сегмента изменяется на целое число

#### экстентов. Блок базы, в другой терминологии страница памяти, это минимальная единица хранения, которой база данных обменивается с диском. Блок базы образуется из нескольких блоков операционной системы. Минимальный размер блока 4 Кбайта. Максимальный -- 64 Кбайт. Существует несколько списков блоков пригодных для записи. Состоит блок из: заголовка, строк данных, списка

#### указателей таблиц, списка указателей строк.