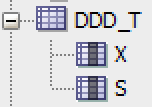
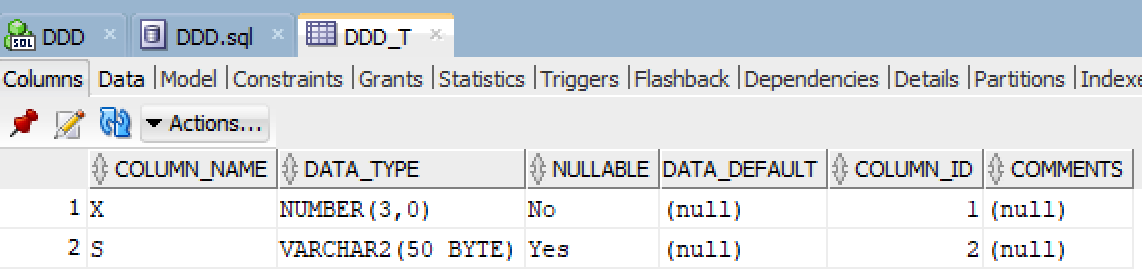
1. В созданный файл MyStart скрипт, создающий таблицу с именем XXX\_t из двух столбцов, первый числовой (используйте тип number c ограничением длины 3), второй символьный (используйте тип varchar2 c ограничением длины 50); первый столбец имеет ограничение целостности первичный ключ.

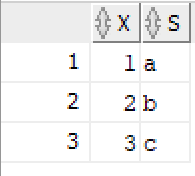
|  |  |
| --- | --- |
| CREATE TABLE DDD\_T( x number(3) primary key, s varchar2(50)); |  |

1. Обновите список объектов (**Refresh**). Найдите в папке Tables созданную таблицу XXX\_t.

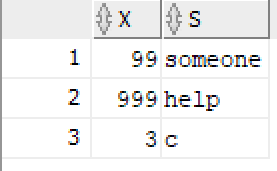
1. Дополните скрипт операторами INSERT, добавляющими 3 строки в таблицу XXX\_t. Выполните операторы INSERT. Выполните оператор COMMIT.

|  |
| --- |
| INSERT ALL  INTO DDD\_T (x, s) VALUES (1, 'a')  INTO DDD\_T (x, s) VALUES (2, 'b')  INTO DDD\_T (x, s) VALUES (3, 'c')  SELECT \* FROM DDD\_T;  COMMIT; |



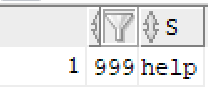
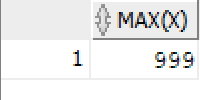
1. Дополните скрипт оператором UPDATЕ, изменяющим 2 строки в таблице XXX\_t. Выполните оператор UPDATЕ. Выполните оператор COMMIT.

|  |
| --- |
| UPDATE DDD\_T SET x = 99, s = 'someone' WHERE x = 1;  UPDATE DDD\_T SET x = 999, s = 'help' WHERE x = 2;  COMMIT; |



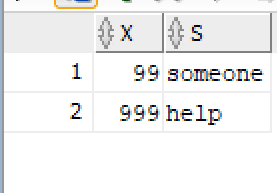
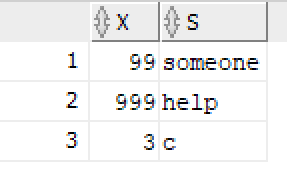
1. Дополните скрипт операторами SELECT (выборка по условию, применение агрегатных функций).

|  |
| --- |
| SELECT \* FROM DDD\_T WHERE s = 'help';  SELECT max(x) FROM DDD\_T; |

1. Дополните скрипт оператором DELELE, удаляющим 1 строку в таблице XXX\_t. Выполните оператор DELETE. Выполните оператор ROLLBACK.

|  |
| --- |
| DELETE FROM DDD\_t WHERE x = 3;  COMMIT;  Rollback; |

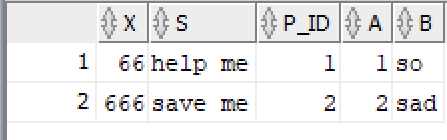
  

1. Создайте таблицу XXX\_t\_child, связанную отношением внешнего ключа с таблицей XXX\_t. Добавьте данные в таблицу XXX\_t\_child.

|  |
| --- |
| ALTER TABLE DDD\_T ADD p\_id int primary key;  CREATE TABLE DDD\_t\_child  (  a number(3),  b varchar2(50),  constraint id\_pk foreign key (a) REFERENCES DDD\_T(p\_id)  );  INSERT ALL  INTO DDD\_t\_child (a,b) VALUES (1, 'so')  INTO DDD\_t\_child (a,b) VALUES (2, 'sad')  SELECT \* FROM dual;  COMMIT; |

1. Дополните скрипт операторами SELECT из обеих таблиц (левое соединение, внутреннее соединение).

|  |
| --- |
| SELECT \* FROM DDD\_T t LEFT OUTER JOIN DDD\_t\_child t1 ON t.p\_id = t1.a;  SELECT \* FROM DDD\_T t RIGHT JOIN DDD\_t\_child t1 ON t.p\_id = t1.a;  SELECT \* FROM DDD\_T t INNER JOIN DDD\_t\_child t1 ON t.p\_id = t1.a; |



1. Дополните скрипт оператором DROP, удаляющим таблицы XXX\_t, XXX\_t\_child.

|  |
| --- |
| DROP TABLE DDD\_T;  DROP TABLE DDD\_t\_child; |

1. Full code

|  |
| --- |
| CREATE TABLE DDD\_T( x number(3) primary key, s varchar(50));  INSERT ALL  INTO DDD\_T (x, s) VALUES (1, 'a')  INTO DDD\_T (x, s) VALUES (2, 'b')  INTO DDD\_T (x, s) VALUES (3, 'c')  SELECT \* FROM dual;  COMMIT;  UPDATE DDD\_T SET x = 99, s = 'someone' WHERE x = 1;  UPDATE DDD\_T SET x = 999, s = 'help' WHERE x = 2;  COMMIT;  SELECT \* FROM DDD\_T WHERE s = 'help';  SELECT max(x) FROM DDD\_T;  DELETE FROM DDD\_t WHERE x = 3;  COMMIT;  Rollback;  DELETE FROM DDD\_T;  ALTER TABLE DDD\_T ADD p\_id int primary key;  CREATE TABLE DDD\_t\_child  (  a number(3),  b varchar2(50),  constraint id\_pk foreign key (a) REFERENCES DDD\_T(p\_id)  );  INSERT ALL  INTO DDD\_T (p\_id, x, s) VALUES (1, 66, 'help me')  INTO DDD\_T (p\_id, x, s) VALUES (2, 666, 'save me')  SELECT \* FROM dual;  COMMIT;  INSERT ALL  INTO DDD\_t\_child (a,b) VALUES (1, 'so')  INTO DDD\_t\_child (a,b) VALUES (2, 'sad')  SELECT \* FROM dual;  COMMIT;  SELECT \* FROM DDD\_T t LEFT OUTER JOIN DDD\_t\_child t1 ON t.p\_id = t1.a;  SELECT \* FROM DDD\_T t RIGHT JOIN DDD\_t\_child t1 ON t.p\_id = t1.a;  SELECT \* FROM DDD\_T t INNER JOIN DDD\_t\_child t1 ON t.p\_id = t1.a;  DROP TABLE DDD\_T;  DROP TABLE DDD\_t\_child; |

1. Что включает в себя администрирование баз данных?
2. Установка и настройка базы данных.
3. Мониторинг и управление производительностью базы данных.
4. Резервное копирование и восстановление данных.
5. Создание и управление пользователями и правами доступа.
6. Обновление и установка патчей базы данных.
7. Управление хранилищем данных.
8. Управление сетевыми соединениями и настройка безопасности
9. Понятия RTO и RPO

RTO (Recovery Time Objective) и RPO (Recovery Point Objective) - это понятия, связанные с восстановлением базы данных. RTO определяет максимальное время восстановления базы данных после сбоя, а RPO определяет максимальный допустимый уровень потерь данных, которые могут произойти в случае сбоя.

1. Основные системные пользователи Oracle

Основные системные пользователи Oracle включают SYS, SYSTEM, DBSNMP и OUTLN. SYS и SYSTEM - это два основных пользователя базы данных Oracle, которые обладают максимальными привилегиями и могут выполнять практически любую операцию в базе данных. DBSNMP - это пользователь, используемый для мониторинга базы данных, а OUTLN - это пользователь, используемый для поддержки определенных функций в базе данных.

1. Как называется системная роль Администратора базы данных

Системная роль Администратора базы данных называется DBA (Database Administrator).

1. Какие есть специальные системные привилегии

Специальные системные привилегии Oracle позволяют пользователям выполнять определенные задачи, которые требуют повышенных привилегий доступа. Некоторые из этих привилегий включают SYSDBA, SYSOPER, AUDIT\_ADMIN, и SELECT\_CATALOG\_ROLE.

1. Понятие экземпляра oracle

Экземпляр Oracle - это процесс базы данных, который запускается и работает в операционной системе. Он управляет памятью, структурами данных и процессами, необходимыми для работы базы данных.

1. Запуска и остановка экземпляра как происходят

Запуск и остановка экземпляра происходят с помощью утилиты управления базой данных Oracle, такой как SQL\*Plus, SQL Developer или Enterprise Manager. Для запуска экземпляра используется команда STARTUP, а для остановки - команда SHUTDOWN.

1. Global Database Name и SID.

Global Database Name и SID - это идентификаторы базы данных Oracle. Global Database Name - это уникальное имя базы данных, которое используется для идентификации базы данных в сети. Он включает имя базы данных и доменное имя. SID - это системное идентификатор экземпляра базы данных, который используется для идентификации экземпляра базы данных в операционной системе. Он может быть раз