

dbrecover pour oracle manuel de l'utilisateur

DBRECOVER pour Oracle Manuel de l'utilisateur 0.5

Aperçu

DBRECOVER pour Oracle est un logiciel de récupération de données Oracle de niveau entreprise. Il peut extraire et récupérer directement les données des fichiers de données (datafile) de la base de données Oracle 8i à 21c, sans avoir à exécuter SQL à travers une instance de base de données Oracle pour récupérer les données. DBRECOVER, développé en Java, n'a pas besoin d'être installé en plus, il peut être utilisé directement après avoir été téléchargé et décompressé.

DBRECOVER utilise une interface graphique intuitive, l'opération est simple et pratique. Les utilisateurs n'ont pas besoin d'apprendre un ensemble de commandes supplémentaires, et ils n'ont pas besoin de comprendre le principe de la structure de données sous-jacente d'Oracle, ils peuvent facilement récupérer les données de la base de données via l'assistant de récupération (Recovery Wizard).

DBRecover for Oracle email: service@parnassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

Database

TABLE: PD.EMP this view only shows some sample data

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30
7902	FORD	ANALYST	7566	03-DEC-1981 00:00:00 AD	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982 00:00:00 AD	1300		10
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30
7902	FORD	ANALYST	7566	03-DEC-1981 00:00:00 AD	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982 00:00:00 AD	1300		10
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30

Version 2009 - Community Edition - 2020.09.25

Copyright © 2012 - 2020 ParnassusData Software, Inc.

<https://www.dbrecover.com>

Pourquoi choisir DBRECOVER ?

Vous vous demandez peut-être si l'utilisation de RMAN, le gestionnaire de récupération traditionnel d'Oracle, n'est pas suffisante pour la sauvegarde et la récupération ? Pourquoi devrions-nous choisir DBRECOVER ? Laissez-moi répondre à vos questions.

Avec la croissance rapide des systèmes IT des entreprises, la capacité des données augmente à un rythme exponentiel. Les DBA d'Oracle sont souvent confrontés à des problèmes tels que l'insuffisance de la capacité de stockage sur disque pour contenir une sauvegarde complète, et le temps moyen de réparation nécessaire pour récupérer les données à partir d'une sauvegarde sur bande dépasse largement les prévisions.

"Pour une base de données, la sauvegarde est primordiale", c'est une maxime que tous les DBA gardent à l'esprit. Cependant, les environnements réels varient énormément, et il est courant que les espaces de sauvegarde des données dans l'environnement de base de données de l'entreprise soient insuffisants, que les dispositifs de stockage achetés ne puissent pas être livrés à court terme, et même que les sauvegardes s'avèrent inutilisables pendant le processus de récupération des données.

Afin de résoudre ces problèmes courants de récupération de données dans le monde réel, le logiciel DBRECOVER exploite pleinement sa compréhension des structures de données internes d'Oracle, des processus de démarrage de base, etc. Il peut gérer des situations où la base de données ne peut pas s'ouvrir correctement, comme la perte d'espace de table SYSTEM, l'opération erronée des tables de dictionnaire de données Oracle, l'incohérence du dictionnaire de données causée par une coupure de courant, etc., même en l'absence totale de sauvegarde. Il peut corriger des erreurs humaines telles que la suppression (Truncate) / suppression (Drop/Delete) / tables de données d'entreprise, et récupérer les données en toute sérénité.

Même les personnes non-DBA qui n'ont eu contact avec la base de données Oracle que pendant quelques jours peuvent facilement utiliser DBRECOVER. Cela est dû à l'installation simple de DBRECOVER et à son interface homme-machine entièrement graphique. Le personnel effectuant la récupération n'a pas besoin de connaissances professionnelles en base de données, n'a pas besoin d'apprendre aucune commande, et n'a certainement pas besoin de comprendre la structure de stockage sous-jacente de la base de données. Il suffit de quelques clics de souris pour récupérer les données en toute sérénité. DBRECOVER a brisé cette limitation qui ne permettait qu'à quelques professionnels d'effectuer des tâches de récupération de base de données, réduisant considérablement le temps de récupération des données suite à un dysfonctionnement de la base de données et réduisant le coût total de la récupération des données pour l'entreprise.

DBRECOVER peut récupérer les données sous deux formes. La méthode traditionnelle d'extraction consiste à extraire complètement les données du fichier de données et à les écrire dans un fichier texte plat, puis à les importer à nouveau dans la base de données à l'aide d'outils tels que SQLLDR. Cette méthode est simple et intuitive, mais nécessite un espace deux fois plus grand que la capacité de données actuelle : l'un est l'espace occupé par les données de texte plat, et l'autre est l'espace nécessaire pour importer les données de texte dans la base de données ; en même temps, en termes de temps, il est nécessaire d'extraire d'abord les données

originales du fichier de données, puis de les importer dans la nouvelle base de données, ce qui nécessite généralement deux fois plus de temps.

Nous recommandons fortement une autre méthode, à savoir le mode DataBridge innovant de DBRECOVER. Cette méthode permet de charger directement les données extraites dans une nouvelle base de données ou une autre base de données disponible via DBRECOVER, évitant ainsi le stockage des données au sol. Par rapport à la méthode traditionnelle, elle permet d'économiser de manière efficace l'espace et le temps nécessaires à la récupération des données.

La technologie ASM (Automatic Storage Management) d'Oracle est de plus en plus adoptée par les entreprises. Par rapport aux systèmes de fichiers traditionnels, les bases de données qui utilisent le stockage ASM ont des performances élevées, supportent les clusters et sont faciles à gérer. Cependant, le problème avec ASM est que sa structure de stockage est trop complexe et difficile à comprendre pour l'utilisateur moyen. Si la structure de données interne d'un Disk Group dans ASM est endommagée et ne peut pas être montée avec succès, les données importantes de l'utilisateur seront "verrouillées" dans cette "boîte noire" ASM. Dans ce cas, il est généralement nécessaire qu'un ingénieur senior de l'usine Oracle qui est familier avec la structure de données interne de ASM arrive sur le site de l'utilisateur pour réparer manuellement la structure interne de ASM ; cependant, l'achat du service sur site de l'usine Oracle est souvent coûteux et long pour l'utilisateur moyen.

C'est parce que les développeurs de DBRECOVER ont une compréhension approfondie de la structure de données interne d'Oracle ASM que DBRECOVER a ajouté une fonction de récupération de données spécifique à ASM.

Actuellement, les fonctions de récupération de données ASM prises en charge par DBRECOVER incluent :

1. Même si le Disk Group ne peut pas être monté normalement, vous pouvez toujours lire directement les métadonnées disponibles sur le disque ASM via DBRECOVER, et copier les fichiers ASM du Disk Group en fonction de ces métadonnées.

2. Même si le Disk Group ne peut pas être monté normalement, vous pouvez toujours lire directement les fichiers de données sur ASM via DBRECOVER, et extraire les données de ces fichiers, en supportant à la fois la méthode d'extraction traditionnelle et le mode DataBridge.

Présentation du logiciel DBRECOVER pour Oracle

DBRECOVER pour Oracle est développé en JAVA, ce qui lui permet de fonctionner sur diverses plateformes, que ce soit sur des plateformes Unix telles que AIX, Solaris, HPUX, sur des plateformes Linux telles que Redhat, Oracle Linux, SUSE, ou sur une plateforme Windows.

Plateformes d'exploitation prises en charge par DBRECOVER :

Nom de la plateforme	Prise en charge
Windows	Oui
AIX	Oui
Solaris Sparc/X86	Oui
Linux x86/64	Oui
HPUX	Oui
MacOS	Oui

Versions de base de données actuellement prises en charge par DBRECOVER : 8i ~ 21C

DBRECOVER intègre l'environnement JAVA nécessaire à son fonctionnement, il n'est donc pas nécessaire d'installer d'autres logiciels JAVA sur Windows/Linux.

Pour l'exécuter sur Windows, double-cliquez sur `start_dbrecover_windows_local_java.bat`

Pour l'exécuter sur Linux, exécutez : `sh start_dbrecover_linux_local_java.sh`

Pour les environnements UNIX tels que AIX/HPUX/Solaris, il est nécessaire que l'utilisateur installe lui-même l'environnement JAVA 8.

Jeux de caractères de base de données pris en charge par DBRECOVER :

Langue	Jeu de caractères	Codage
Chinois Simplifié/Traditionnel	ZHS16GBK	GBK
Chinois Simplifié/Traditionnel	ZHS16DBCS	CP935
Chinois Simplifié/Traditionnel	ZHT16BIG5	BIG5
Chinois Simplifié/Traditionnel	ZHT16DBCS	CP937
Chinois Simplifié/Traditionnel	ZHT16HKSCS	CP950
Chinois Simplifié/Traditionnel	ZHS16CGB231280	GB2312
Chinois Simplifié/Traditionnel	ZHS32GB18030	GB18030
Japonais	JA16SJIS	SJIS
Japonais	JA16EUC	EUC_JP
Japonais	JA16DBCS	CP939
Coréen	KO16MSWIN949	MS649
Coréen	KO16KSC5601	EUC_KR
Coréen	KO16DBCS	CP933
Français	WE8MSWIN1252	CP1252
Français	WE8ISO8859P15	ISO8859_15
Français	WE8PC850	CP850
Français	WE8EBCDIC1148	CP1148
Français	WE8ISO8859P1	ISO8859_1
Français	WE8PC863	CP863
Français	WE8EBCDIC1047	CP1047
Français	WE8EBCDIC1147	CP1147
Allemand	WE8MSWIN1252	CP1252
Allemand	WE8ISO8859P15	ISO8859_15
Allemand	WE8PC850	CP850
Allemand	WE8EBCDIC1141	CP1141
Allemand	WE8ISO8859P1	ISO8859
Italien	WE8MSWIN1252	CP1252
Italien	WE8ISO8859P15	ISO8859_15

Italien	WE8PC850	CP850
Italien	WE8EBCDIC1144	CP1144
Thaï	TH8TISASCII	CP874
Thaï	TH8TISEBCDIC	TIS620
Arabe	AR8MSWIN1256	CP1256
Arabe	AR8ISO8859P6	ISO8859_6
Arabe	AR8ADOS720	CP864
Espagnol	WE8MSWIN1252	CP1252
Espagnol	WE8ISO8859P1	ISO8859_1
Espagnol	WE8PC850	CP850
Espagnol	WE8EBCDIC1047	CP1047
Portugais	WE8MSWIN1252	CP1252
Portugais	WE8ISO8859P1	ISO8859_1
Portugais	WE8PC850	CP850
Portugais	WE8EBCDIC1047	CP1047
Portugais	WE8ISO8859P15	ISO8859_15
Portugais	WE8PC860	CP860

Types de stockage de table pris en charge par DBRECOVER :

Type de stockage de table	Prise en charge
Table de grappes Cluster Table	OUI
Table organisée par index, partitionnée ou non	NON
Table de tas ordinaire, partitionnée ou non	OUI
Table de tas ordinaire avec compression de base activée	NON
Table de tas ordinaire avec compression avancée activée	NON
Table de tas ordinaire avec compression de colonnes mixtes activée	NON
Table de tas ordinaire avec chiffrement activé	NON
Table avec colonnes virtuelles	NON
Lignes enchaînées, lignes migrées	OUI

Points à noter : Pour les colonnes virtuelles et les colonnes par défaut optimisées en 11g, l'extraction des données peut ne pas poser de problème, mais les champs correspondants

seront perdus. Ces deux caractéristiques sont nouvelles après la version 11g et sont moins utilisées.

Types de données de colonnes pris en charge par DBRECOVER :

Type de données	Prise en charge
BFILE	Non
XML binaire	Non
BINARY_DOUBLE	Oui
BINARY_FLOAT	Oui
BLOB	Oui
CHAR	Oui
CLOB et NCLOB	Oui
Collections (y compris VARRAYS et tables imbriquées)	Non
Date	Oui
INTERVAL DAY TO SECOND	Oui
INTERVAL YEAR TO MONTH	Oui
LOBs stockés en tant que SecureFiles	Oui
LONG	Oui
LONG RAW	Oui
Types de données multimédia (y compris Spatial, Image, et Oracle Text)	Non
NCHAR	Oui
Number	Oui
NVARCHAR2	Oui
RAW	Oui
ROWID, UROWID	Oui
TIMESTAMP	Oui
TIMESTAMP WITH LOCAL TIMEZONE	Oui
TIMESTAMP WITH TIMEZONE	Oui
Types définis par l'utilisateur	Non
VARCHAR2 et VARCHAR	Oui
XMLType stocké en tant que CLOB	Non

XMLType stocké en tant qu'objet relationnel	Non
---	-----

Prise en charge de ASM par DBRECOVER :

Fonction	Prise en charge
Extraction directe des données depuis ASM, sans nécessité de copie sur le système de fichiers	OUI
Copie de fichiers de données depuis ASM	OUI

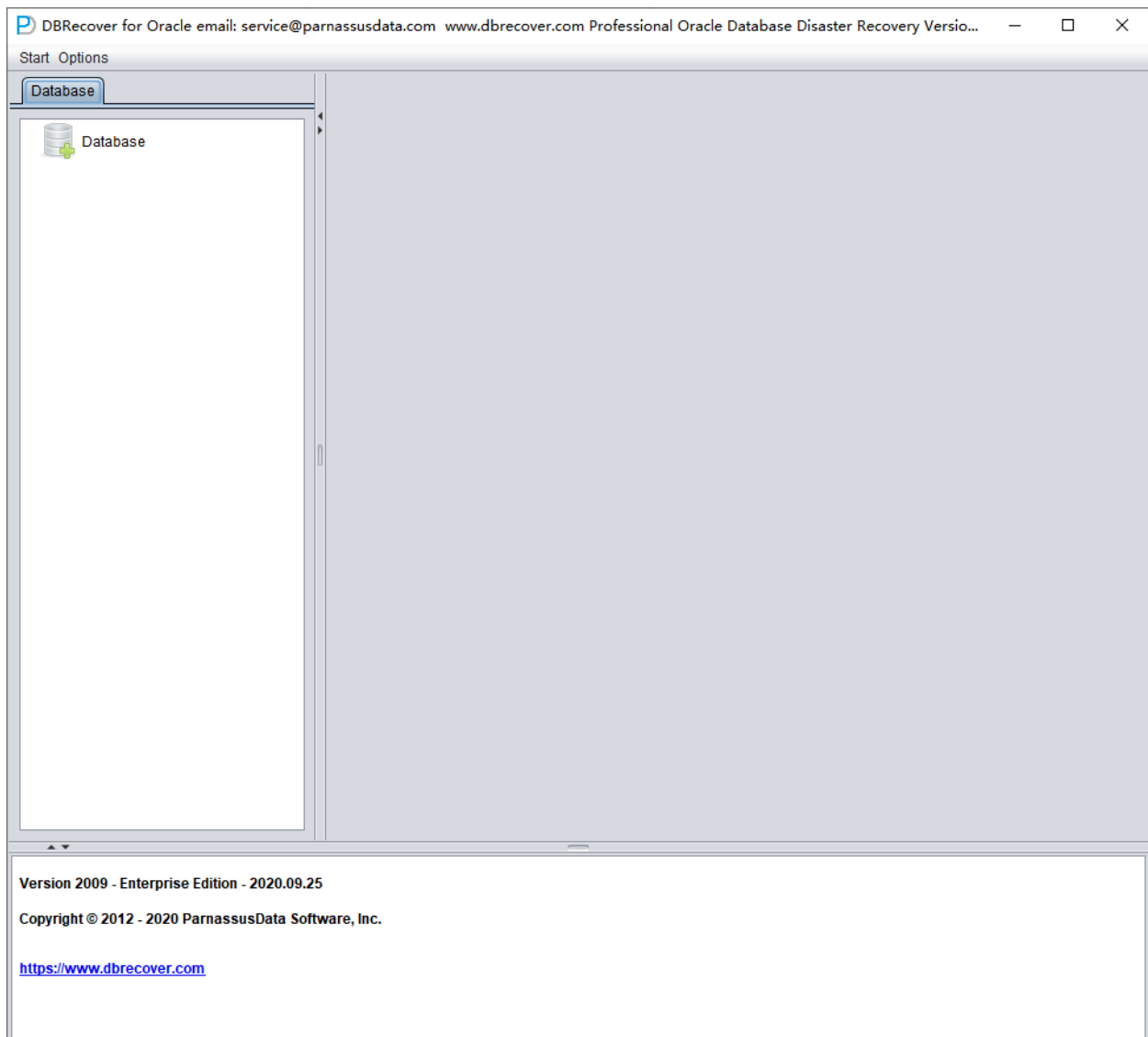
Installation et démarrage de DBRECOVER

Comme DBRECOVER est un logiciel entièrement vert développé en JAVA, il n'est pas nécessaire de l'installer séparément. Les utilisateurs n'ont qu'à décompresser le fichier ZIP du logiciel téléchargé pour récupérer les données.

Pour l'exécuter sur Windows, double-cliquez sur `start_dbrecover_windows_local_java.bat`

Dans un environnement Linux, il peut être utilisé sur l'interface graphique locale ou via des outils graphiques à distance tels que Xmanager/VNC.

1. Assurez-vous que vous pouvez ouvrir le petit programme d'horloge graphique `xclock`.
2. Exécutez dans le répertoire de décompression du logiciel : `sh start_dbrecover_linux_local_java.sh`



Dans un environnement AIX/HPUX/Solaris, il peut être utilisé sur l'interface graphique locale ou via des outils graphiques à distance tels que Xmanager/VNC.

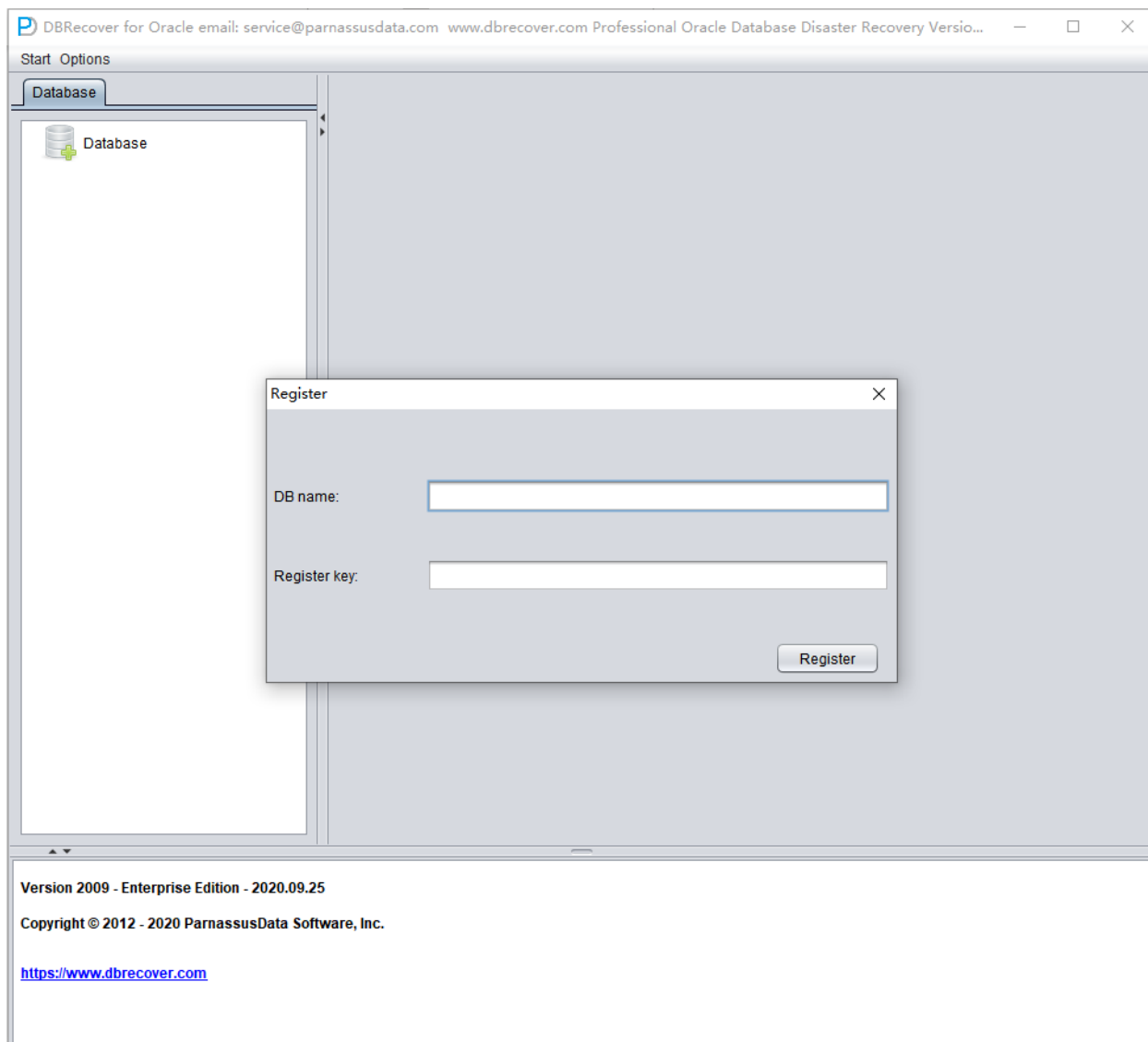
1. Assurez-vous que l'environnement JAVA 8 correspondant est installé et confirmez avec la commande : `java -version`
2. Assurez-vous que vous pouvez ouvrir le petit programme d'horloge graphique `xclock`
3. Exécutez dans le répertoire de décompression du logiciel : `sh start_dbrecover.sh`

Enregistrement de la licence DBRECOVER

DBRECOVER For Oracle est un logiciel commercial. La version communautaire de DBRECOVER est disponible pour les tests et l'apprentissage des utilisateurs.

Nous n'offrons actuellement qu'un seul type de licence, à savoir la licence d'entreprise. Vous pouvez obtenir des informations d'achat sur le site <https://www.dbrecover.com/>.

Après avoir obtenu la clé de licence, l'utilisateur peut s'enregistrer lui-même dans le logiciel Register. La méthode spécifique d'utilisation est la suivante :



Dans la barre de menu Help => Register, entrez le nom de la base de données et la clé selon les informations qui vous ont été envoyées après l'achat, puis cliquez sur le bouton Register. Une fois l'enregistrement terminé, le redémarrage ultérieur de DBRECOVER détectera automatiquement les informations d'enregistrement de la licence et il n'est pas nécessaire de se réenregistrer.

Les informations d'enregistrement réussies peuvent être trouvées dans Help=>About :



Présentation de l'utilisation de DBRECOVER basée sur différents scénarios de récupération de base de données Oracle

Scénario de récupération 1 ORACLE Les fichiers de données endommagés rendent la base de données incapable de s'ouvrir OPEN

La base de données de production de l'entreprise A fonctionne en mode non-archivé tout au long de l'année, fait occasionnellement une sauvegarde logique EXP, mais ne fait jamais de sauvegarde physique. Un jour, après un redémarrage dû à une panne de courant, la base de données n'a pas pu s'ouvrir normalement pour utilisation. Après vérification, il a été constaté que l'espace de table SYSTEM était gravement endommagé. À ce stade, vous pouvez utiliser DBRECOVER pour transférer rapidement les données de la base de données endommagée vers une nouvelle base de données afin de reprendre rapidement les opérations.

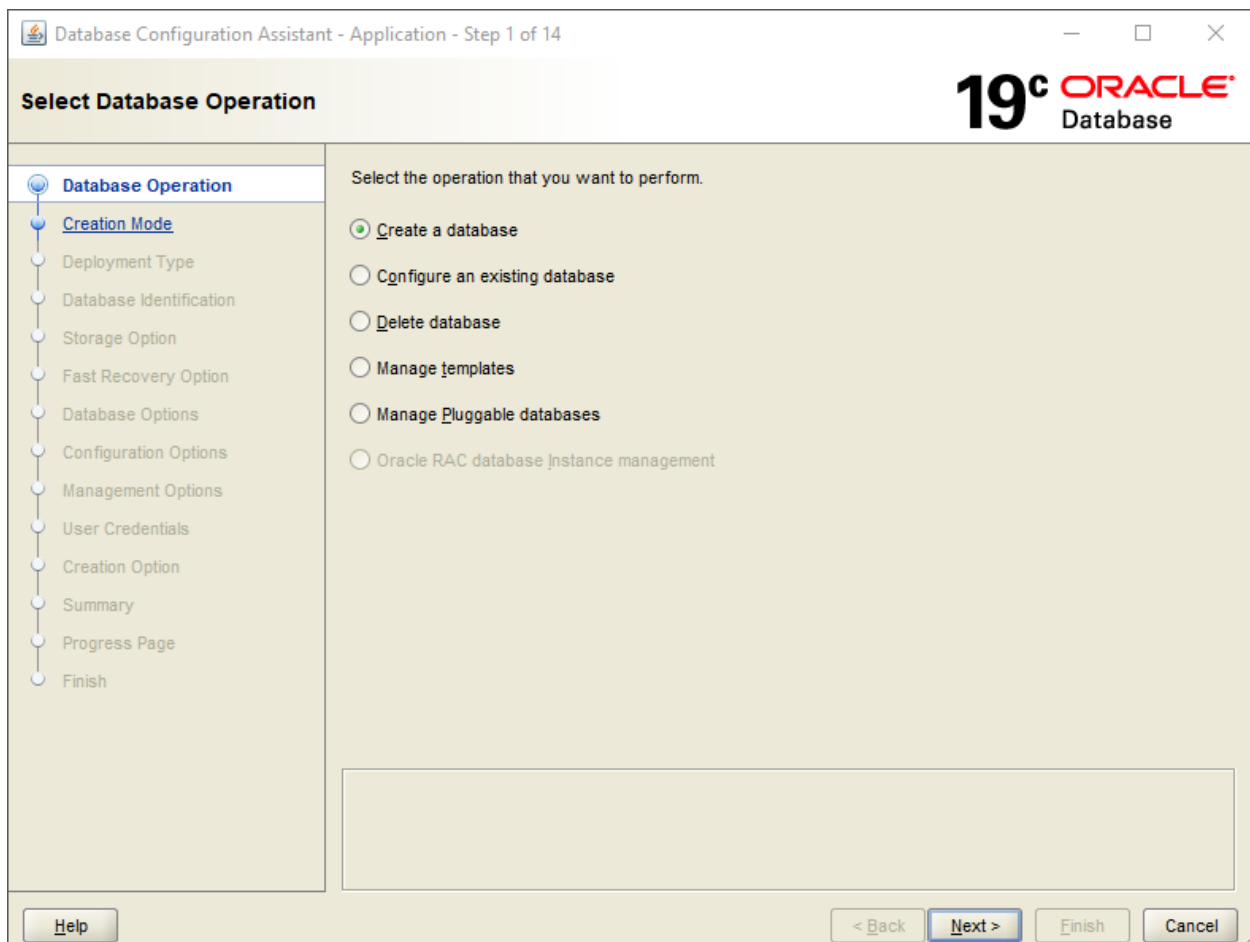
Des situations similaires à ce scénario, telles que celles où la base de données ne peut pas s'ouvrir en raison d'erreurs ORA-01194 ORA-01110 ORA-01033 ORA-01115 ORA-00368 ORA-00600 kcbzib_kcrsds_1 ORA-00333 ORA-01113 ORA-01122 ORA-27027, etc., peuvent toutes utiliser la méthode utilisée dans ce scénario de récupération pour tenter de récupérer les données.

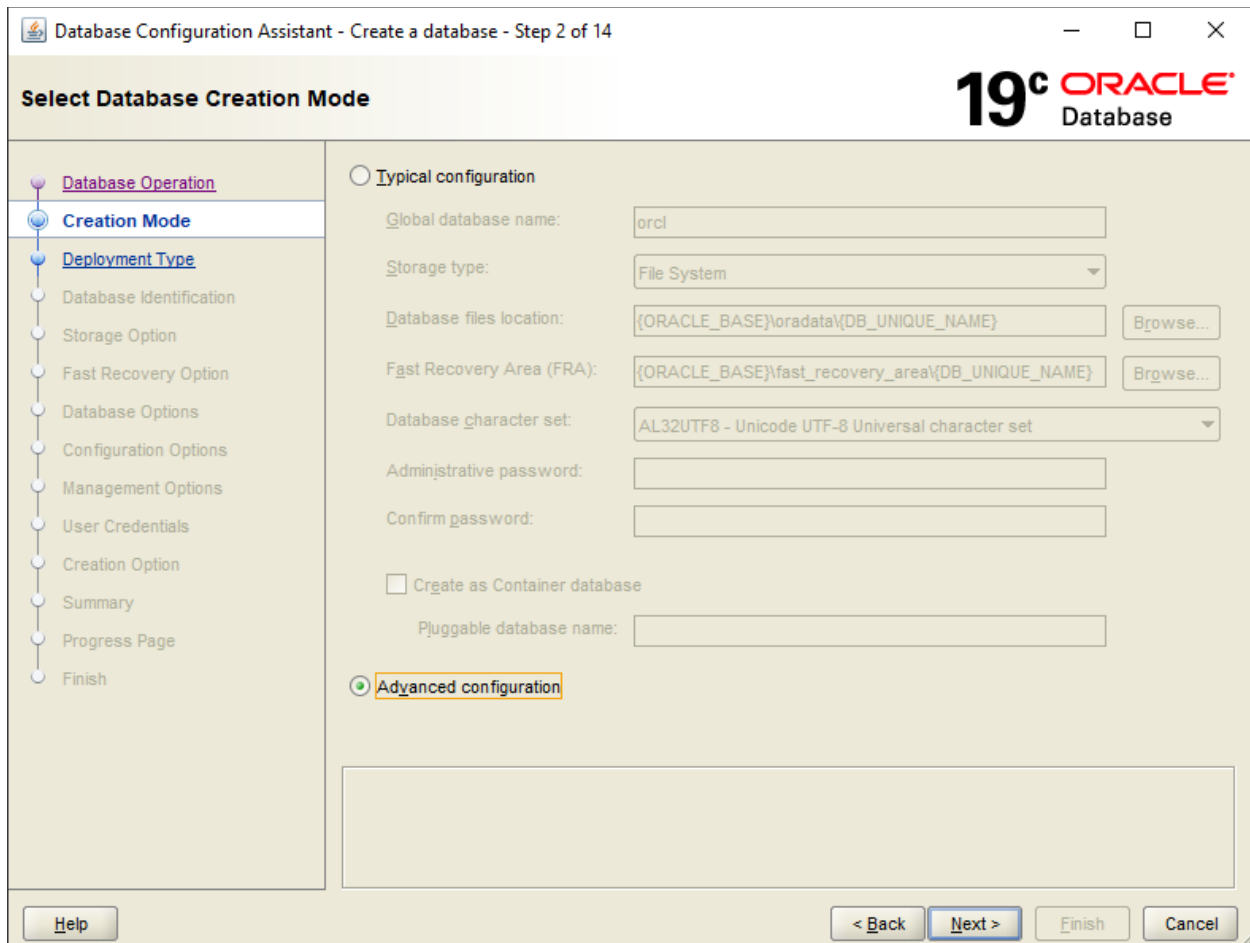
Les étapes simplifiées sont les suivantes :

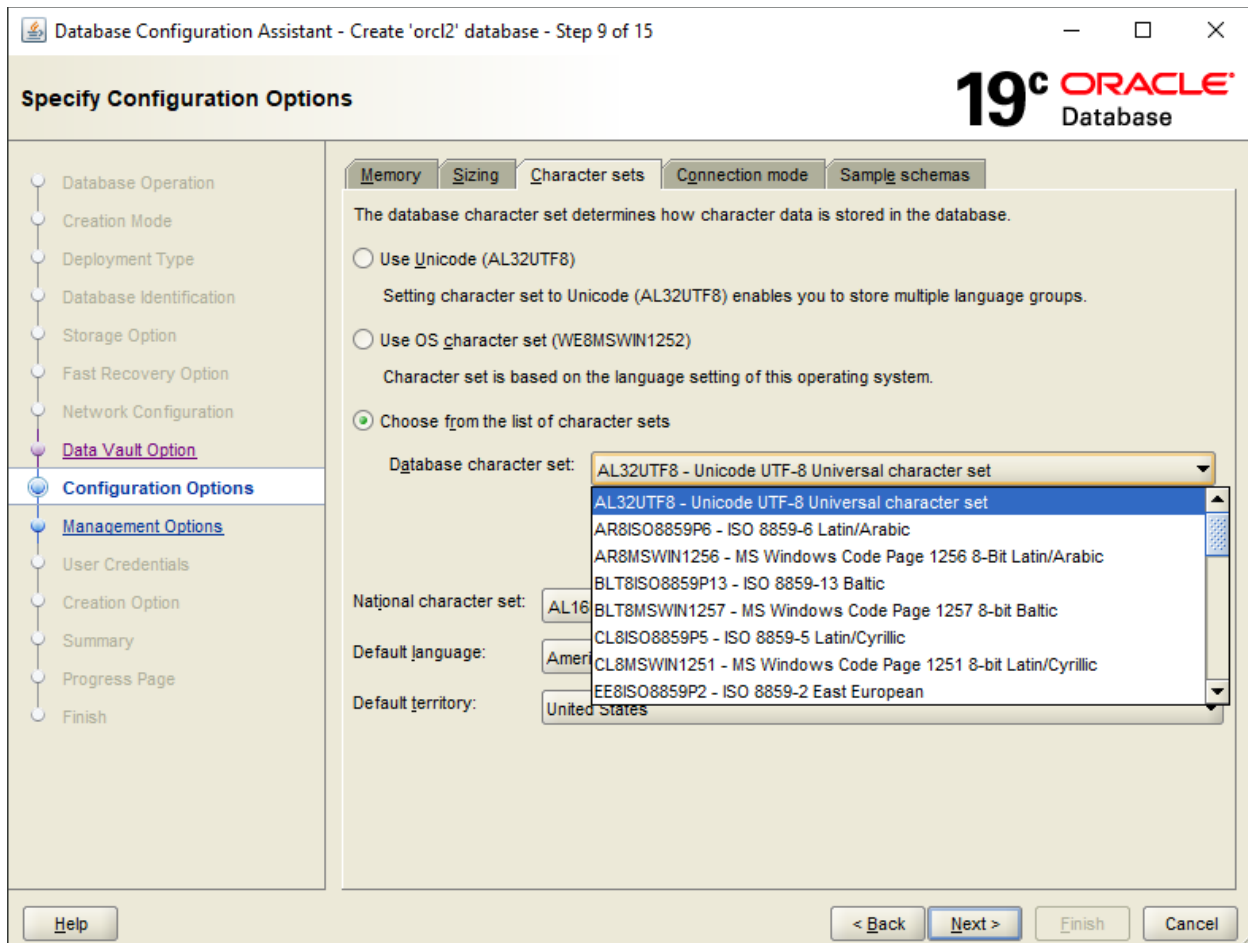
1. Utilisez dbca pour créer une nouvelle base de données ORACLE, notez que le jeu de caractères doit correspondre à la base de données endommagée
2. Créez des utilisateurs et des espaces de table correspondants dans la nouvelle base de données, il est recommandé de donner temporairement le rôle DBA à ces utilisateurs
3. Démarrez le programme d'écoute (LISTENER), assurez-vous que le service de base de données est enregistré à l'écoute
4. Démarrez DBRECOVER en mode dictionnaire et chargez tous les fichiers de données de la base de données d'origine endommagée
5. Dans DBRECOVER, cliquez sur le nom d'utilisateur à récupérer, cliquez avec le bouton droit et choisissez le pont de données
6. Dans l'interface du pont de données, cliquez sur l'icône plus, ajoutez les informations de connexion de la nouvelle base de données (Connection)
7. Cliquez sur Data Bridge pour commencer le travail de transfert, attendez que toutes les tables sous SCHEMA soient transférées vers le SCHEMA de destination de la base de

données de destination

8. Sélectionnez le SCHEMA correspondant, cliquez avec le bouton droit et choisissez la fonction EXPORTDDL, sélectionnez le type d'objet à récupérer, puis cliquez sur EXPORT
9. Basé sur le fichier DDL SQL généré par EXPORTDDL, exécutez-le manuellement dans le SCHEMA de destination de la base de données de destination







// Démarrer le programme d'écoute (LISTENER), assurez-vous que le service de base de données est enregistré à l'écoute

```
C:\Users\testenv>lsnrctl status
```

```
LSNRCTL for 64-bit Windows: Version 11.2.0.1.0 - Production on 12-MAY-2023 10:01:48
```

```
Copyright (c) 1991, 2010, Oracle. All rights reserved.
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=DESKTOP-testenv)(PORT=1521)))
```

```
STATUS of the LISTENER
```

```
-----
```

```
Alias LISTENER
```

```
Version TNSLSNR for 64-bit Windows: Version 11.2.0.1.0 - Production
```

```
Start Date 12-MAY-2023 10:00:49
```

```
Uptime 0 days 0 hr. 0 min. 59 sec
```

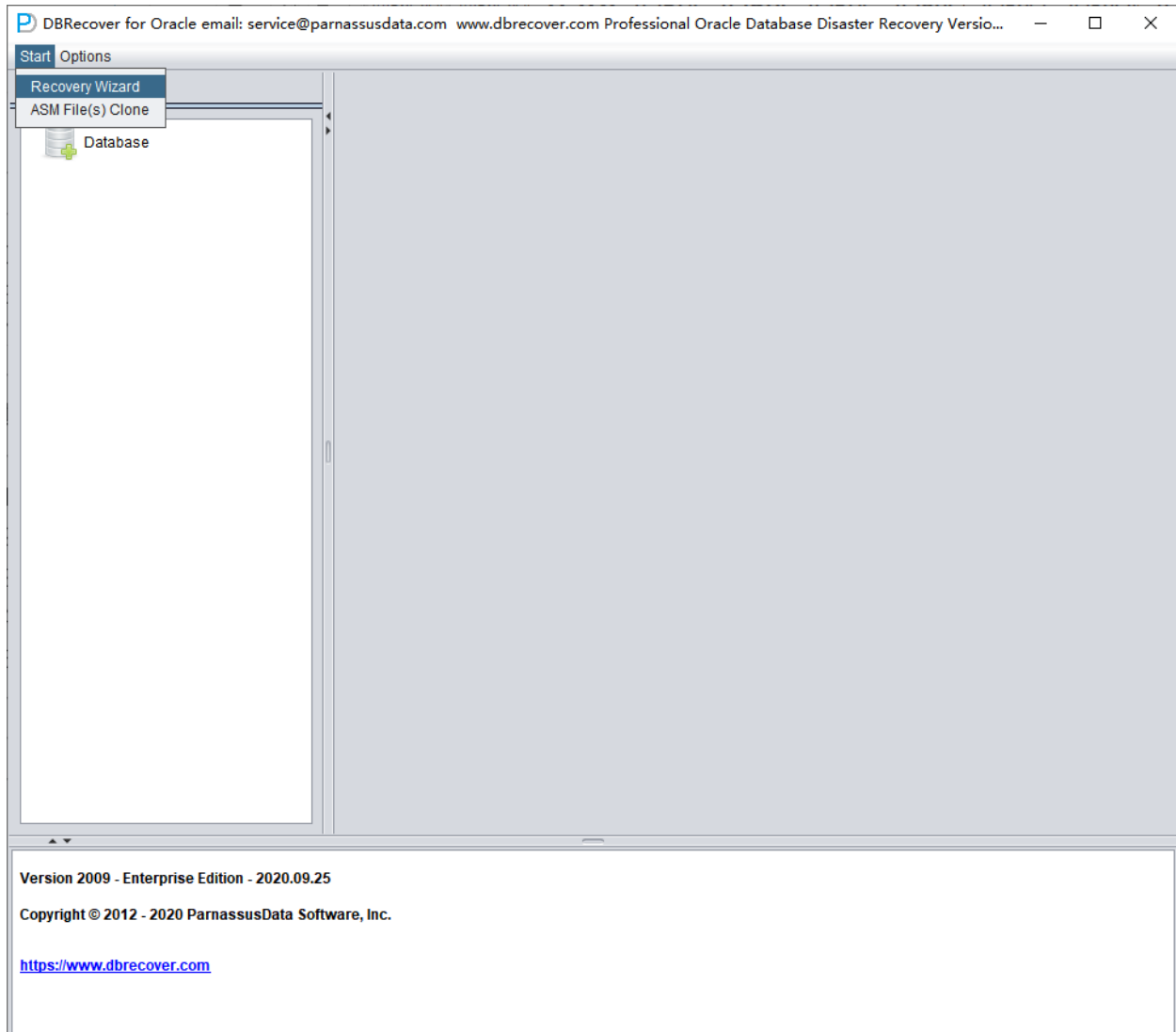
```
Trace Level off
Security ON: Local OS Authentication
SNMP OFF
Listener Parameter File D:\app\testenv\product\11.2.0\dbhome_2\network\admin\listener.ora
Listener Log File d:\app\testenv\diag\tnslsnr\DESKTOP-testenv\listener>alert\log.xml
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=DESKTOP-testenv)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROC1521ipc)))
Services Summary...
Service "CLRExtProc" has 1 instance(s).
Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "ORCL1XDB" has 1 instance(s).
Instance "orcl1", status READY, has 1 handler(s) for this service...
Service "ORCLXDB" has 1 instance(s).
Instance "orcl", status READY, has 1 handler(s) for this service...
Service "orcl" has 1 instance(s).
Instance "orcl", status READY, has 1 handler(s) for this service...
Service "orcl1" has 1 instance(s).
Instance "orcl1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

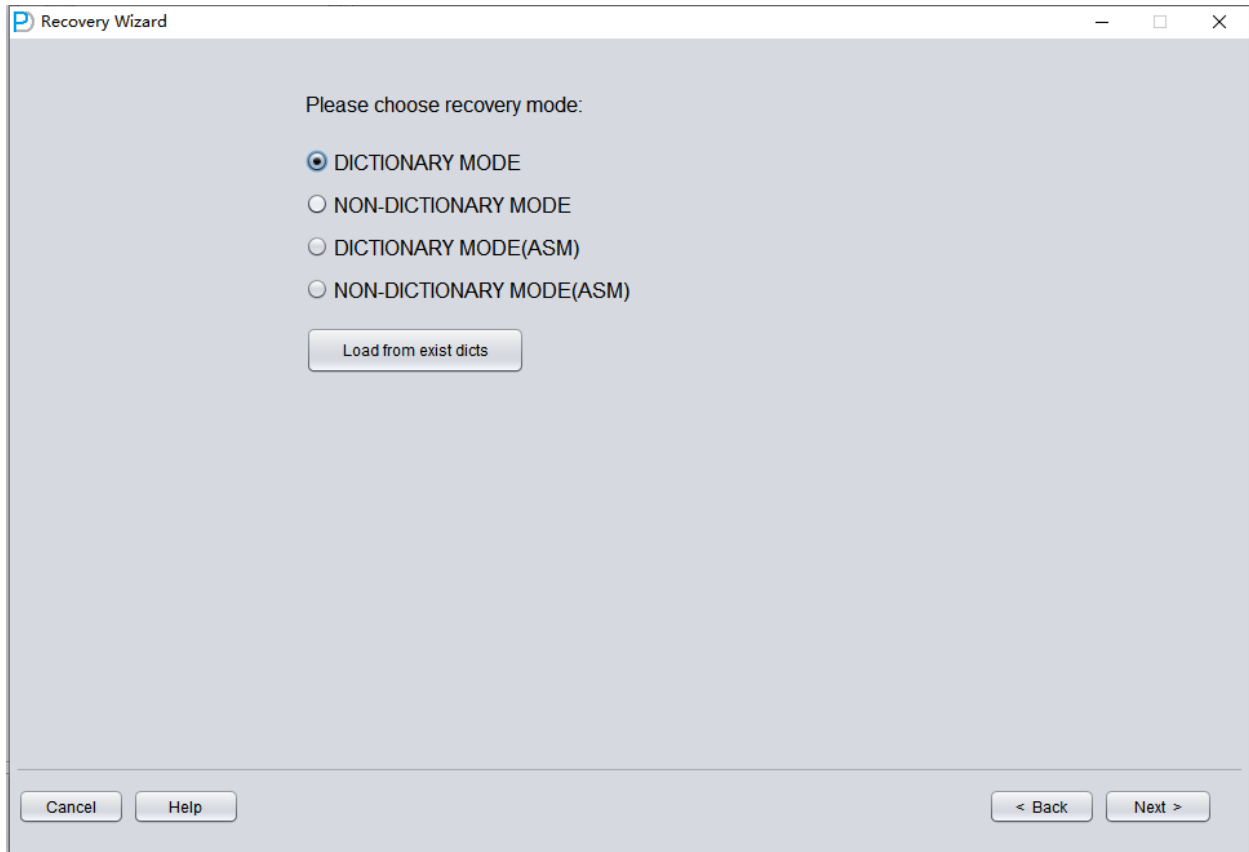
```
// Créez des utilisateurs et des espaces de table correspondants dans la nouvelle base de données, il est recommandé de donner temporairement le rôle DBA à ces utilisateurs
```

```
set ORACLE_SID=ORCL1
sqlplus / as sysdba
SQL> create user pd identified by oracle;
User created.
SQL> grant dba to pd;
Grant succeeded.
SQL> create tablespace pdtbs datafile size 500M autoextend on next 100M;
Tablespace created.
SQL> alter user pd default tablespace pdtbs;
User altered.
```

Démarrer DBRECOVER, et sélectionner Outils => Assistant de récupération

Cliquez sur Suivant





Dans l'étape suivante, nous devons choisir le bon ordre des octets ENDIAN ; puisque les fichiers de données ORACLE utilisent différents formats d'ordre des octets Endian sur différentes plateformes d'exploitation, la liste des correspondances entre l'ordre des octets et la plateforme est la suivante :

plateforme	endian
Solaris[tm] OE (32-bit)	Big
Solaris[tm] OE (64-bit)	Big
Microsoft Windows IA (32-bit)	Little
Linux IA (32-bit)	Little
Systèmes basés sur AIX (64-bit)	Big
HP-UX (64-bit)	Big
HP Tru64 UNIX	Little

HP-UX IA (64-bit)	Big
Linux IA (64-bit)	Little
HP Open VMS	Little
Microsoft Windows IA (64-bit)	Little
IBM zSeries Based Linux	Big
Linux x86 64-bit	Little
Apple Mac OS	Big
Microsoft Windows x86 64-bit	Little
Solaris Operating System (x86)	Little
IBM Power Based Linux	Big
HP IA Open VMS	Little
Solaris Operating System (x86-64)	Little
Apple Mac OS (x86-64)	Little

Il suffit de noter que les plateformes Windows et Linux les plus couramment utilisées sont toutes deux en Little Endian, donc aucun réglage n'est nécessaire, il suffit de conserver la valeur par défaut.

Sur les plateformes de petits ordinateurs, y compris les systèmes basés sur AIX (64-bit) et HP-UX (64-bit), on utilise le grand ordre des octets Big Endian, donc ici il faut choisir Big Endian.

Remarque : Si vos fichiers de données ont été générés sur AIX (c'est-à-dire en Big Endian) et que vous les avez copiés sur un serveur Windows pour faciliter la récupération des données avec DBRECOVER, vous devez toujours choisir le format Big Endian d'origine.

Ici, comme nous devons récupérer les fichiers de la base de données Oracle sur la plateforme Linux x86-64, nous choisissons Endian comme Little.

Cliquez sur Suivant

Recovery Wizard

Endian: Little Endian

DB Character Set: From dictionary

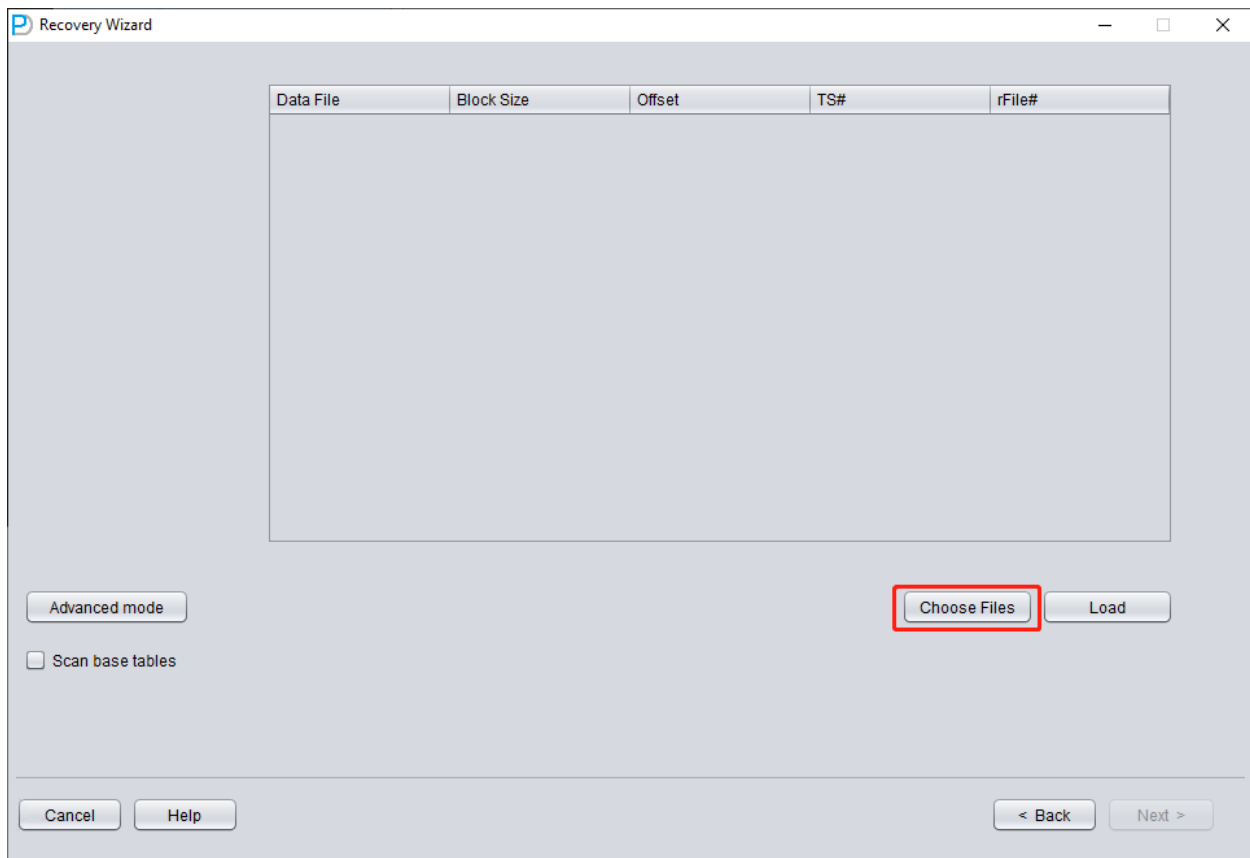
DB National Character Set: From dictionary

Block Size: 8192

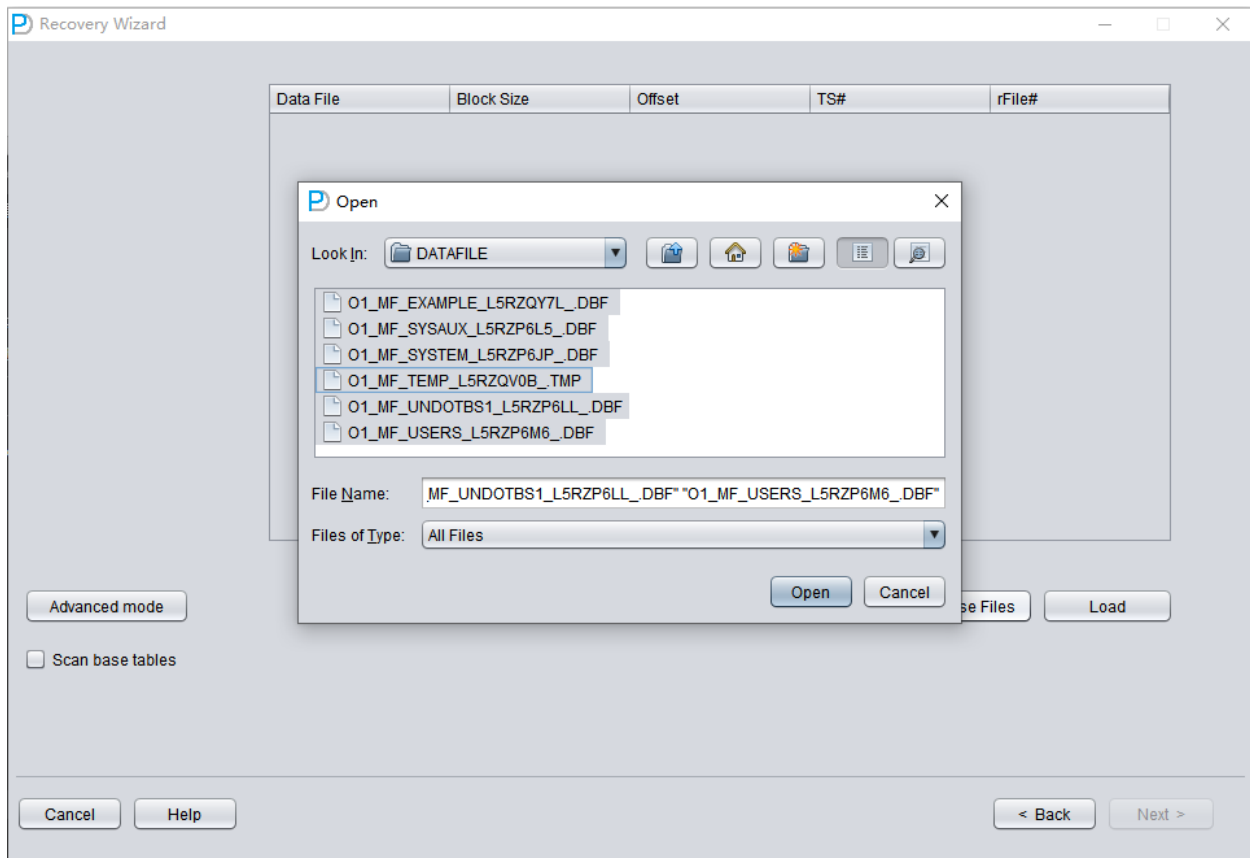
Offset: 0

DB Version: auto detect

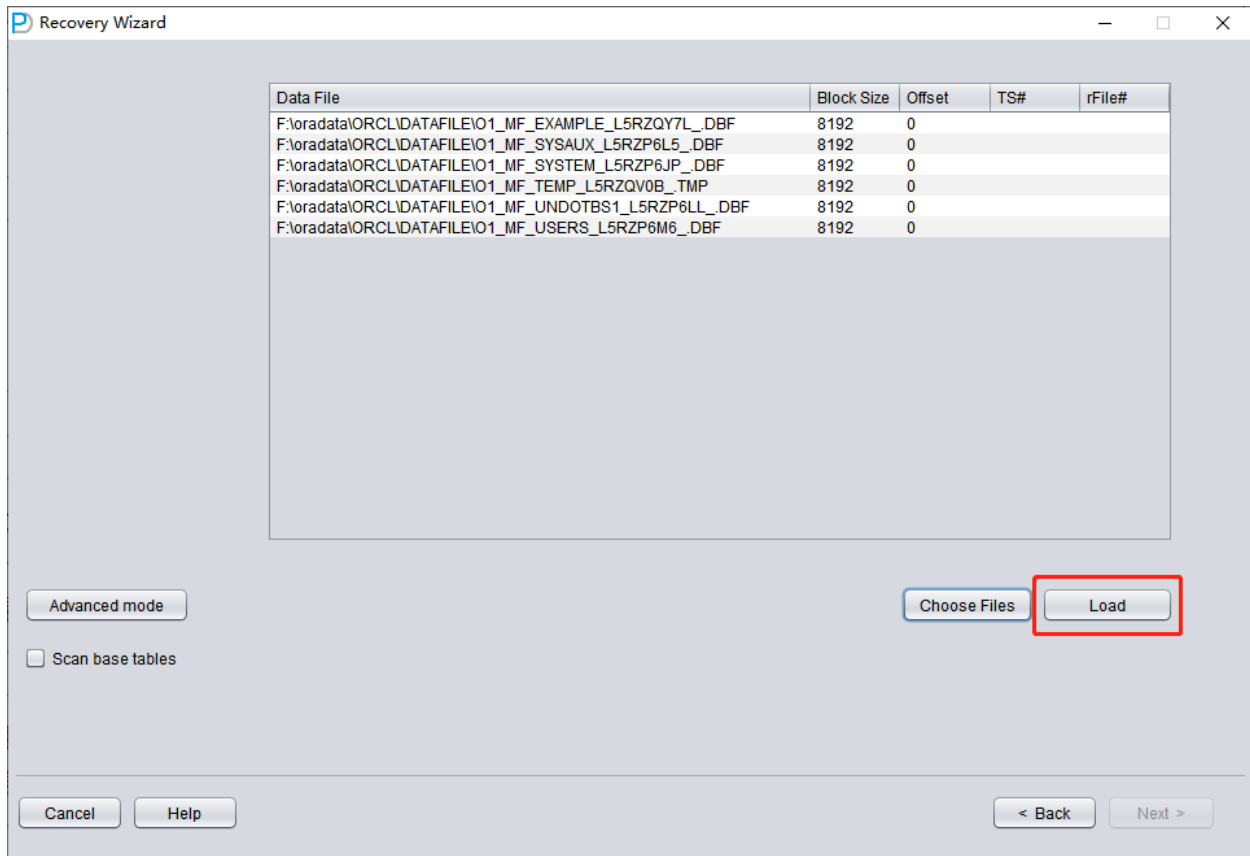
Cancel Help < Back Next >



Cliquez sur Choose Files, généralement nous recommandons si la base de données n'est pas grande, alors sélectionnez tous les fichiers de données de cette base de données ; Si votre base de données est très grande, et que vous savez sur quels fichiers de données se trouvent vos tables de données, alors vous pouvez simplement choisir les fichiers de données de l'espace de table SYSTEM (obligatoire !) et les fichiers de données où se trouvent les données.



Notez que l'interface Choose prend en charge les opérations de clavier telles que Ctrl + A et Shift.



Notez : une fois que tous les fichiers de données ont été ajoutés, si vous ne comprenez pas les autres paramètres de cette interface, alors laissez-les tous par défaut, pas besoin de modifier !

Ensuite, il faut spécifier la taille du bloc, c'est-à-dire la taille du bloc de données ORACLE, pour les fichiers de données spécifiés. Ici, modifiez selon la situation réelle, par exemple, si votre DB_BLOCK_SIZE est de 8K, mais que certains espaces de table spécifient 16K comme taille de bloc de données, alors il suffit de modifier le BLOCK_SIZE pour ces fichiers de données qui ne sont pas de 8k.

Il n'est pas nécessaire de spécifier OFFSET ici pour une utilisation avec un système de fichiers ordinaire. Le paramètre OFFSET est principalement destiné aux scénarios où les fichiers de données sont stockés sur des dispositifs nus, par exemple, si le LV basé sur VG ordinaire sur AIX est utilisé comme fichier de données, alors il y a un OFFSET de 4k qui doit être spécifié ici.

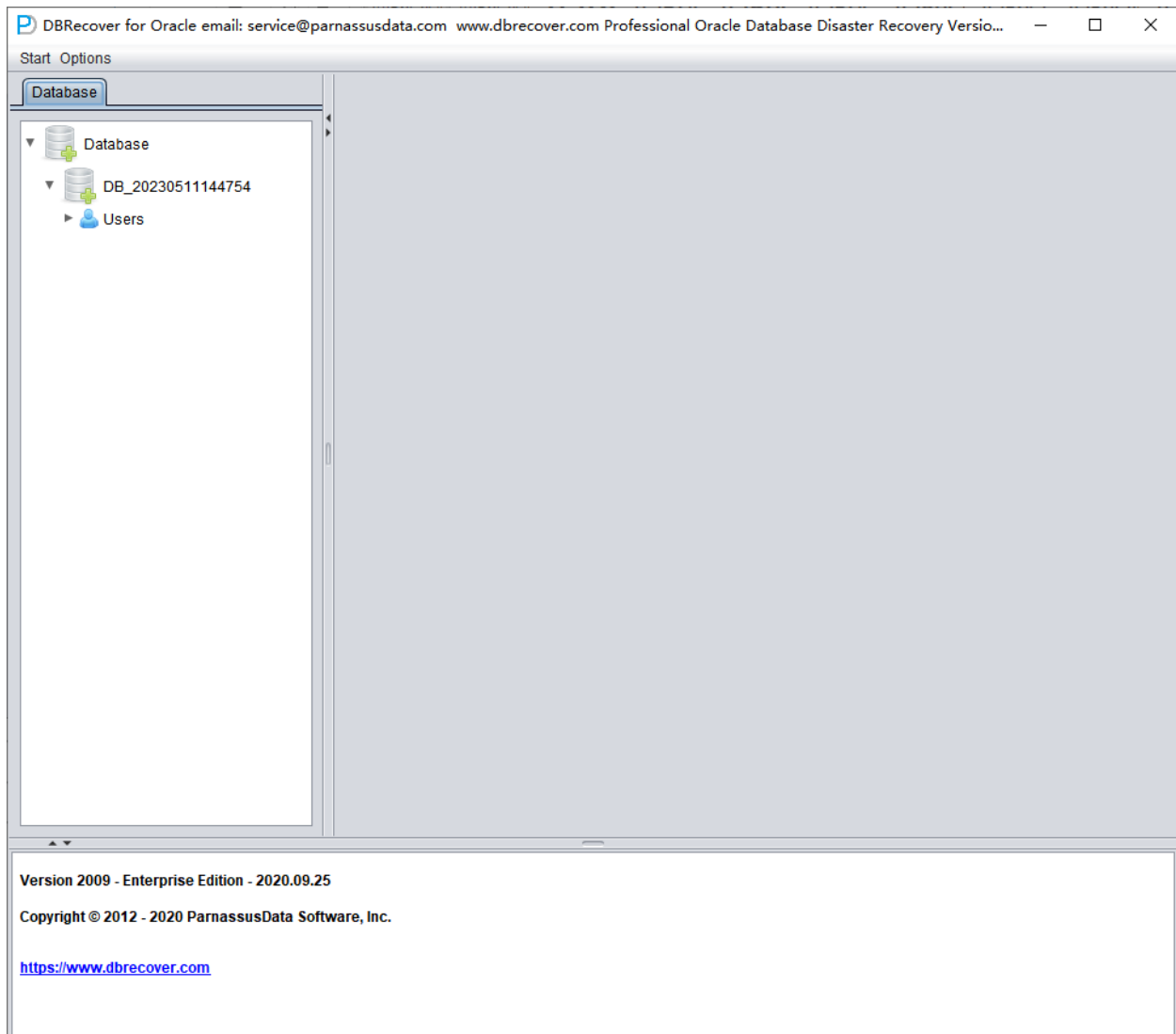
Si vous utilisez justement des fichiers de données sur des dispositifs nus, et que vous ne savez pas combien est OFFSET ? Vous pouvez utiliser l'outil dbfsize inclus avec \$ORACLE_HOME/bin pour le vérifier :

```
$ dbfsize /dev/lv_control_01
Database file: /dev/lv_control_01
Database file type: raw device without 4K starting offset
Database file size: 334 16384 byte blocks
```

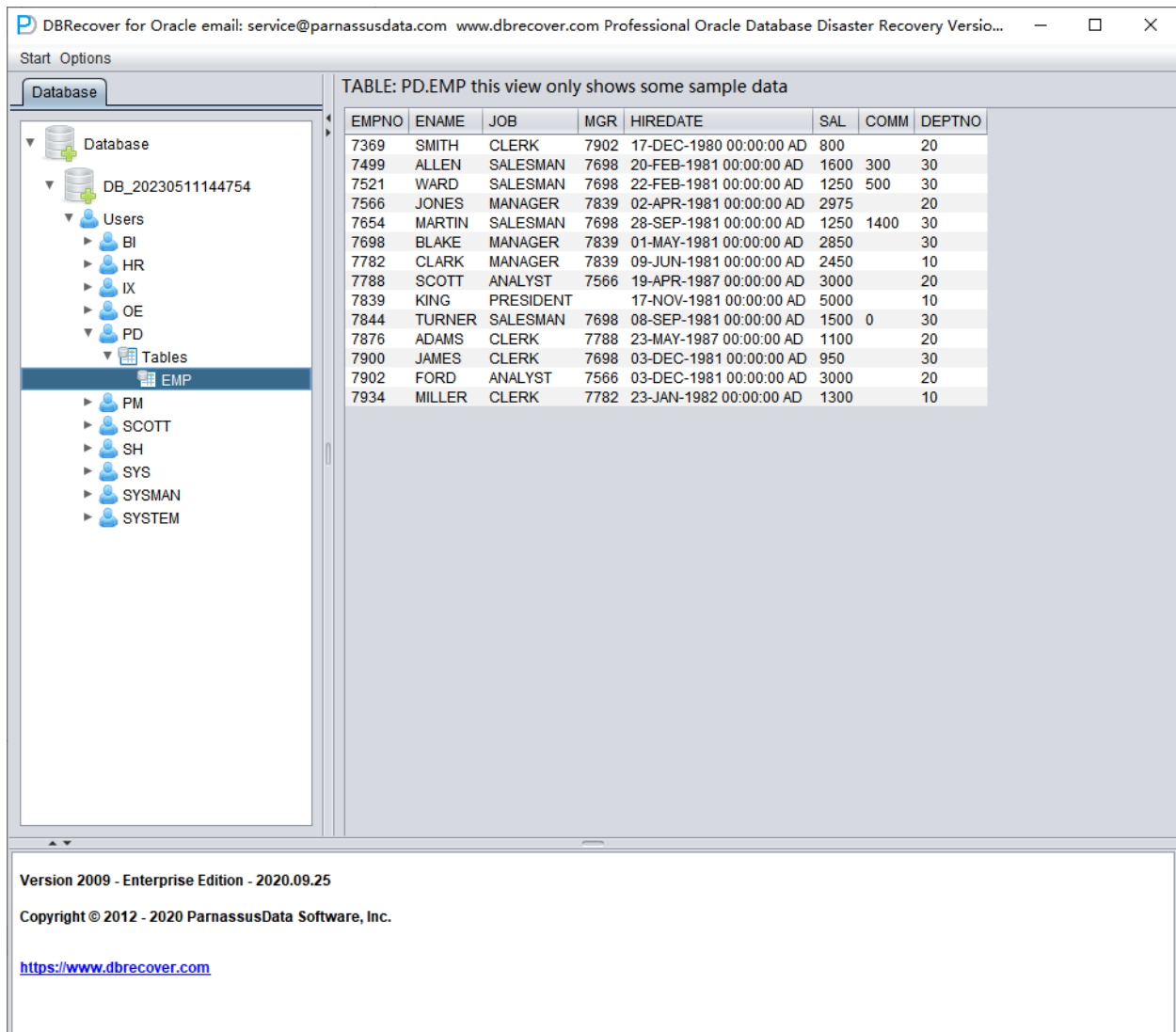
Dans ce scénario, tous les fichiers de données ont une taille de bloc BLOCK SIZE de 8K et sont basés sur un système de fichiers, donc il n'y a pas d'OFFSET, cliquez sur Load

Pendant la phase de chargement, DBRECOVER lira les informations du dictionnaire de données ORACLE à partir de l'espace de table SYSTEM et créera un dictionnaire de données dans son propre Derby, ce qui permet à DBRECOVER d'analyser toutes sortes de données dans la base de données ORACLE.

Une fois le chargement terminé, un arbre regroupé par utilisateur de base de données apparaît sur le côté gauche de l'interface DBRECOVER :



Sélectionnez une table que vous souhaitez récupérer, double-cliquez pour afficher les données :



Sans l'achat d'une licence logicielle, nous pouvons examiner si DBRECOVER peut récupérer suffisamment de données en examinant les tables de données, en extrayant au moins dix mille lignes de données et en vérifiant le nombre de lignes récupérables.

DBRecover for Oracle email: service@parnassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

Database

TABLE: PD.EMP this view only shows some sample data

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30
7902	FORD	ANALYST	7566	03-DEC-1981 00:00:00 AD	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982 00:00:00 AD	1300		10
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30

Version 2009 - Enterprise Edition - 2020.09.25

Copyright © 2012 - 2020 ParnassusData Software, Inc.

<https://www.dbrecover.com>

Après avoir sélectionné la table, faites un clic droit sur UNLOAD, cela exportera les données de la table sous forme de texte :

DBRecover for Oracle

i Unload successfully!

File path: C:\Users\Administrator\Desktop\dbrecover-for-oracle2009\prmd\data\parnassus_dbinfo_DB_20230519122351\pd.emp.dat

Unloaded row count: 10000

Elapsed time(seconds): 0.0

For Community Edition, row limitation is 10,000.

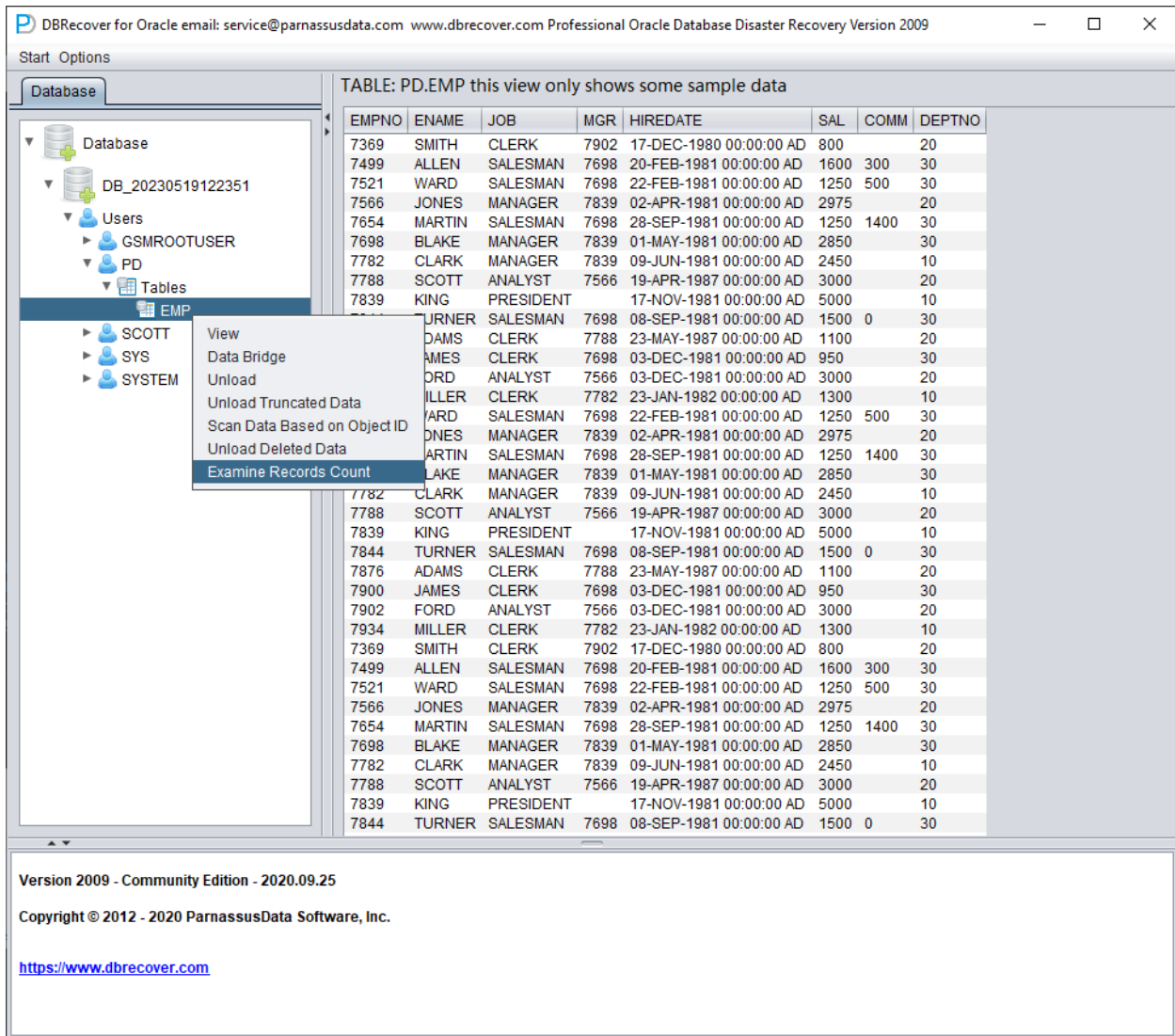
For Enterprise Edition, there is no row limitation.

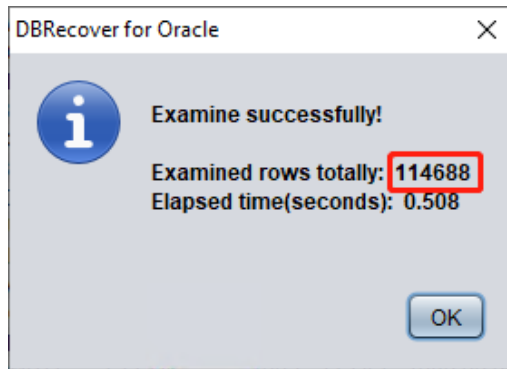
If you need to recover more data, please contact service@parnassusdata.com

OK

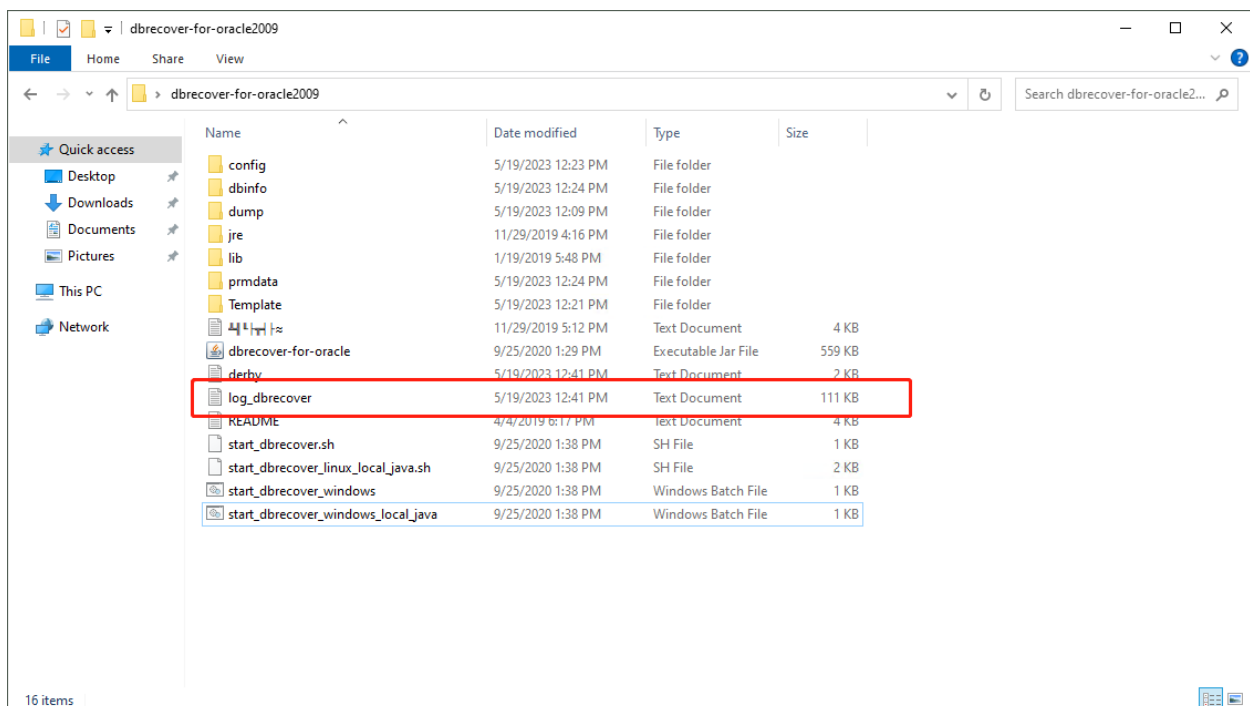
Sans enregistrement de licence logicielle, une seule table peut extraire jusqu'à dix mille lignes de données.

Pour les tables contenant plus de dix mille lignes de données, la fonction de vérification du nombre de lignes récupérables peut être utilisée pour une vérification supplémentaire. Sélectionnez la table à vérifier, faites un clic droit sur EXAMINE RECORDS COUNT :





Oracle a introduit la fonction de collecte automatique de statistiques à partir de 10g, nous pouvons utiliser cette fonction pour consulter les informations statistiques historiques de la table, y compris le nombre de lignes. Dans le mode dictionnaire, chaque fois que nous exécutons des opérations de visualisation, d'extraction, de vérification, etc. sur une table, certaines informations sur cette table sont enregistrées dans le fichier journal log_dbrecover.txt. Le fichier journal est stocké dans le répertoire du logiciel :



```

log_dbrecover - Notepad
File Edit Format View Help
TABLE SYS.TYPE$ 5889 rows unloaded
TABLE SYS.COLLECTION$ 1385 rows unloaded
TABLE SYS.ATTRIBUTE$ 15376 rows unloaded
TABLE SYS.LOBFRAG$ 25 rows unloaded
TABLE SYS.LOBCOMPPART$ 0 rows unloaded
TABLE SYS.TS$ 6 rows unloaded
Warning can be ignored: insert prm_tables_collection rows number is 2206
Warning can be ignored: delete SYS_NC000$ & SYS_C00 & Virtual Column for col$ rows number is 1178
Warning can be ignored: delete SYS_STU SYS_ST5 Column for col$ rows number is 0
Warning can be ignored: delete BIN$ recyclebin object for obj$ rows number is 0
created view pd_tab_col
the manual path for tabpart$ is ./manual/sys.tabpart$.dat
the manual load tabpart$.dat not exists, using default :./prmdata/parnassus_dbinfo_DB_20230519125028/./sys.tabpart$.dat
the manual path for tabsubpart$ is ./manual/sys.tabsubpart$.dat
the manual load tabsubpart$.dat not exists, using default :./prmdata/parnassus_dbinfo_DB_20230519125028/./sys.tabsubpart$.dat
the manual path for lob$ is ./manual/sys.lob$.dat
the manual load lob$.dat not exists, using default :./prmdata/parnassus_dbinfo_DB_20230519125028/./sys.lob$.dat
the manual path for ind$ is ./manual/sys.ind$.dat
the manual load ind$.dat not exists, using default :./prmdata/parnassus_dbinfo_DB_20230519125028/./sys.ind$.dat
the manual path for lobfrag$ is ./manual/sys.lobfrag$.dat
the manual load lobfrag$.dat not exists, using default :./prmdata/parnassus_dbinfo_DB_20230519125028/./sys.lobfrag$.dat
Use default path to load sys.indpart$.dat
Use default path to load sys.indsubpart$.dat
Database character set is AL32UTF8
Database national character set is AL16UTF16
Current character set for decoding is UTF8
Current national character set for decoding is UTF16

For Community Edition, row limitation is 10,000.
If you need to recover more data, please contact service@parnassusdata.com

object information user#:106 object_name: EMP object_id:74042 data_object_id:74042 object_type:2
table information object_id:74042 data_object_id:74042 ts#:4 rfile#:7 block#:386 rowcnt:114688 blkcnt:751 analyzetime:2023-05-19 12:41:29.0
TABLE PD.EMP 666 rows unloaded

```

Informations du journal :

```

object information user#:106 object_name: EMP object_id:74042 data_object_id:74042 object_type:2
table information object_id:74042 data_object_id:74042 ts#:4 rfile#:7 block#:386 rowcnt:114688
blkcnt:751 analyzetime:2023-05-19 12:41:29.0
TABLE PD.EMP 666 rows unloaded

```

Beaucoup d'informations utiles apparaissent dans le journal :

object_id numéro d'objet	74042
data_object_id numéro d'objet de données	74042
ts# numéro d'espace de table	4
rfile# numéro de fichier relatif où se trouve l'en-tête de la table	7
block# numéro de bloc de données où se trouve l'en-tête de la table	386
rowcnt nombre de lignes enregistrées dans les informations statistiques (les informations statistiques sont des estimations)	114688
blkcnt nombre total de blocs de cette table	751

analyzetime heure de collecte des informations statistiques	2023-05-19 12:41:29.0
---	--------------------------

En général, l'erreur des informations statistiques ne dépasse pas 10%, donc nous pouvons comparer le résultat du compte de lignes à partir d'ici. Par exemple, le rowcnt ici est de 114688 (pour les tables de moins de 1 million de lignes, l'erreur des informations statistiques est très petite), et le résultat de l'EXAMINE est de 114688 lignes, ce qui peut vérifier l'authenticité de ce résultat.

En fonction de leurs besoins, les utilisateurs peuvent effectuer le contrôle ci-dessus pour chaque table de données importante. Nous recommandons aux utilisateurs de vérifier complètement l'intégrité des données récupérables avant d'acheter une licence logicielle.

Une fois ces contrôles effectués, nous commençons le transfert de données de pont de SCHEMA au niveau de l'utilisateur, sélectionnez le nom d'utilisateur à récupérer, faites un clic droit sur Data Bridge

DBRecover for Oracle email: service@parnassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

Database

- Database
 - DB_20230519125028
 - Users
 - GSMROOTUSER
 - PD
 - Data Bridge
 - Export DDL
 - SCOTT
 - SYS
 - SYSTEM

TABLE: PD.EMP this view only shows some sample data

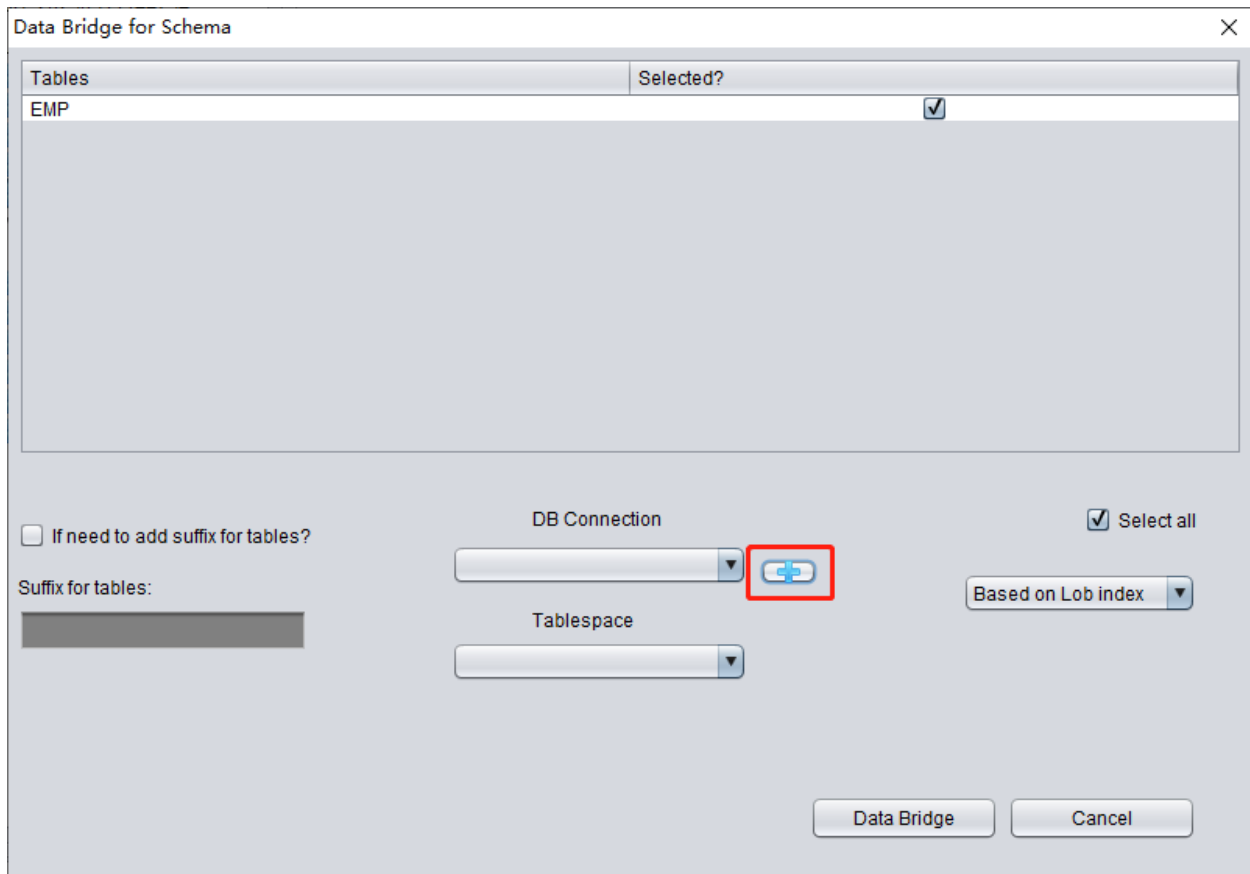
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30
7902	FORD	ANALYST	7566	03-DEC-1981 00:00:00 AD	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982 00:00:00 AD	1300		10
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-1987 00:00:00 AD	1100		20
7900	JAMES	CLERK	7698	03-DEC-1981 00:00:00 AD	950		30
7902	FORD	ANALYST	7566	03-DEC-1981 00:00:00 AD	3000		20
7934	MILLER	CLERK	7782	23-JAN-1982 00:00:00 AD	1300		10
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-1981 00:00:00 AD	2850		30
7782	CLARK	MANAGER	7839	09-JUN-1981 00:00:00 AD	2450		10
7788	SCOTT	ANALYST	7566	19-APR-1987 00:00:00 AD	3000		20
7839	KING	PRESIDENT		17-NOV-1981 00:00:00 AD	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-1981 00:00:00 AD	1500	0	30

Version 2009 - Community Edition - 2020.09.25

Copyright © 2012 - 2020 ParnassusData Software, Inc.

<https://www.dbrecover.com>

Dans l'interface de transfert de données au niveau SCHEMA, cliquez sur le bouton "+", ajoutez les informations de connexion de la base de données cible :

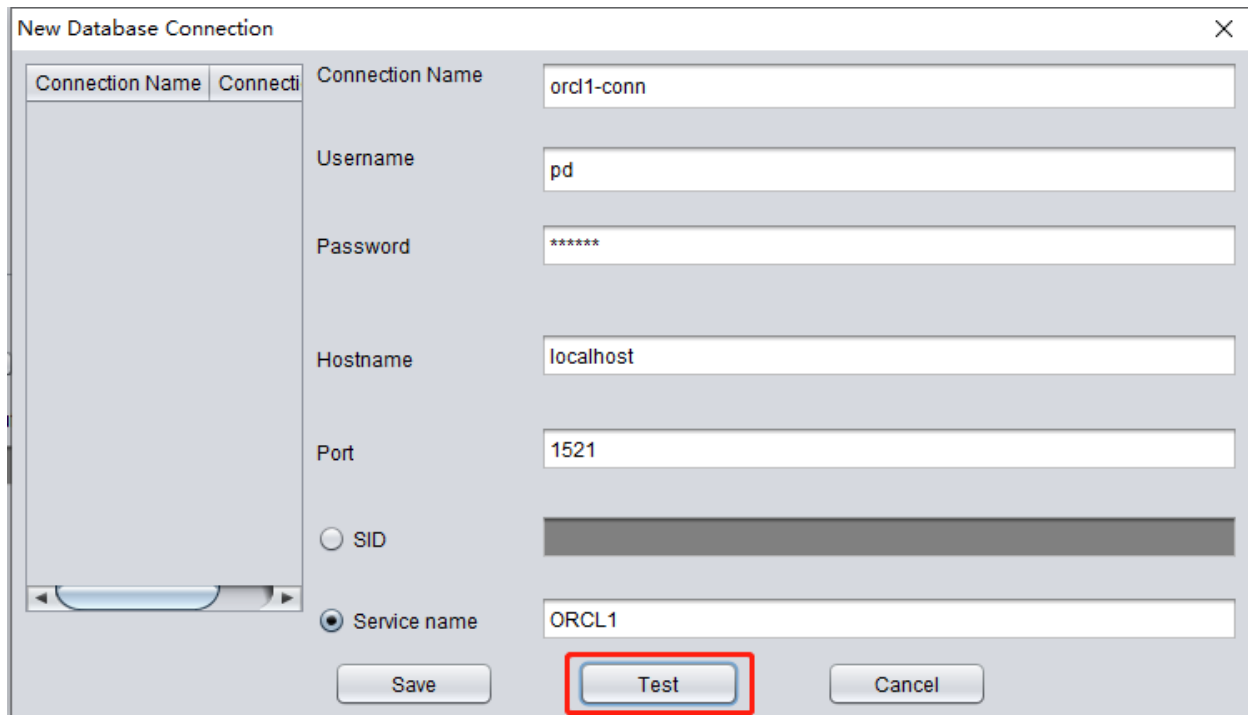


Entrez les informations de connexion de la nouvelle base de données créée précédemment, ici nous utilisons l'utilisateur PD.

Remarque : Le logiciel DBRECOVER ne transférera les données qu'à l'utilisateur dans les informations de connexion de la base de données, c'est-à-dire que si vous saisissez PD ici, les données seront transférées à PD. Les clients doivent ici suivre le principe simple du un à un, c'est-à-dire si un utilisateur de base de données à récupérer, par exemple EAS, doit créer un utilisateur EAS et son espace de table dans la base de données cible et attribuer les droits nécessaires (rôle DBA), et saisir EAS dans ce lien de base de données, pour s'assurer que les données sont transférées à EAS, PD ici n'est qu'un exemple. Si le client souhaite récupérer plusieurs utilisateurs de base de données, tels que EAS, MES, NC001, etc., il doit créer ces comptes et leurs espaces de table dans la base de données cible en conséquence, et attribuer les droits nécessaires (rôle DBA), puis créer plusieurs informations de connexion de base de données (DB Connection) dans DBRECVOER, et spécifier les informations de connexion de

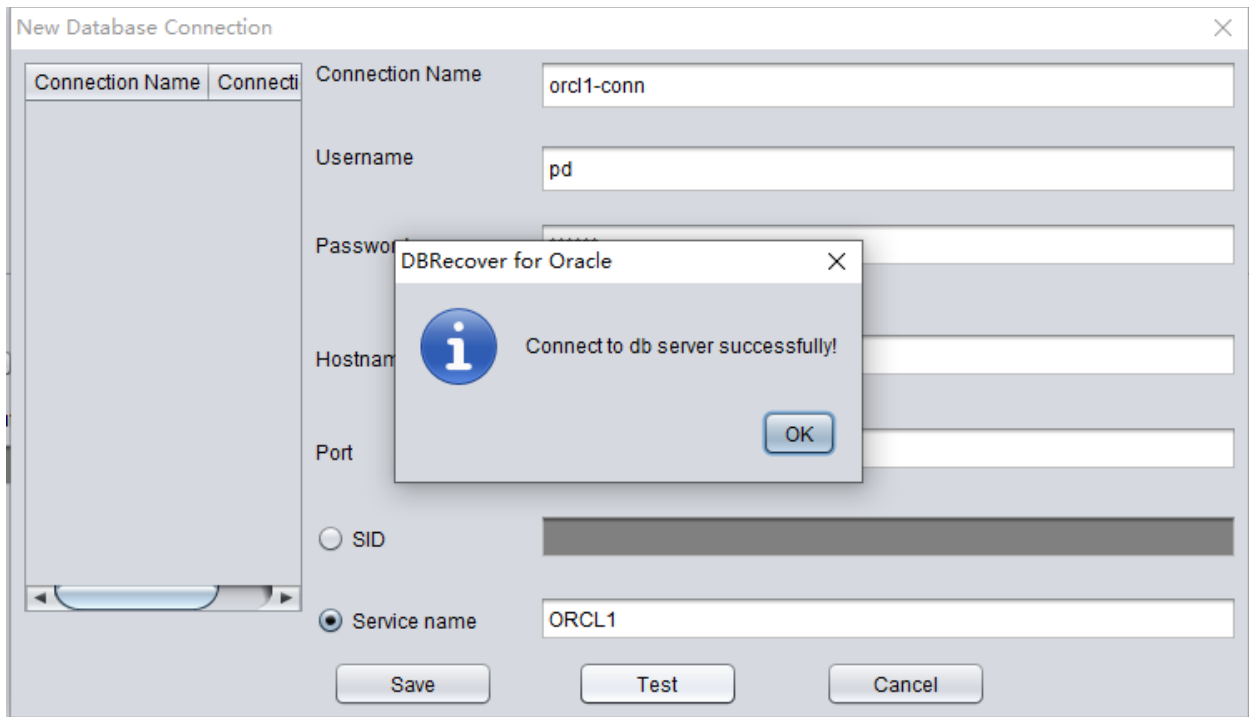
base de données correspondantes (DB Connection) lors de la transmission du SCHEMA d'un utilisateur spécifique.

Cliquez sur TEST pour tester la disponibilité de la connexion à la base de données cible :

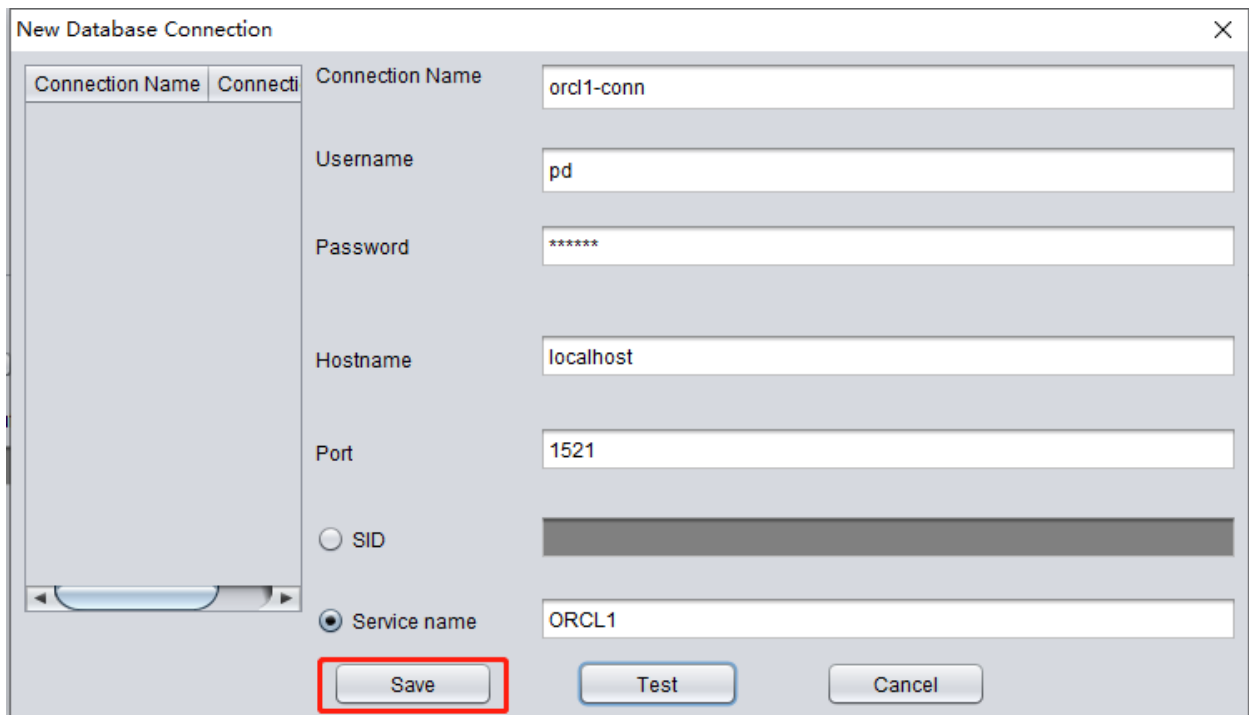


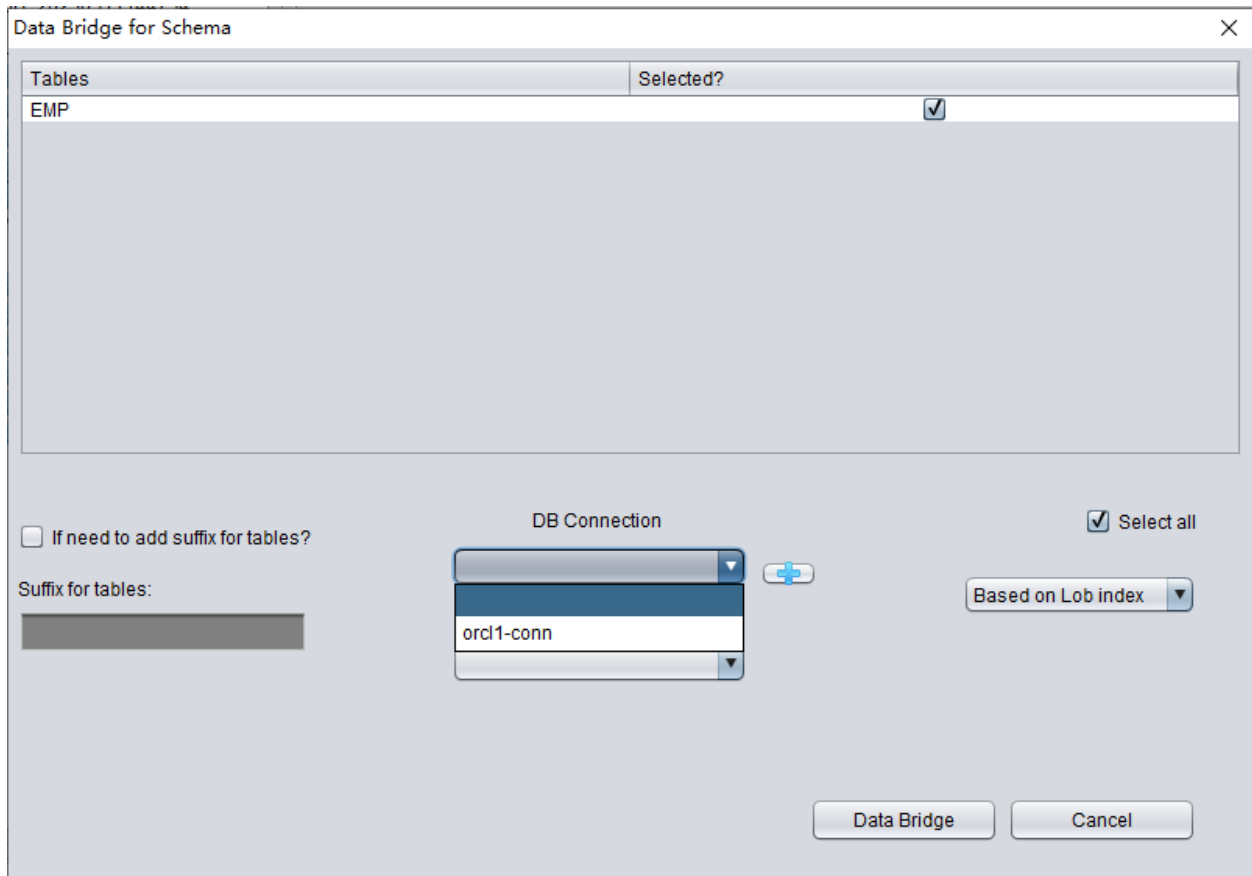
The image shows a 'New Database Connection' dialog box with the following fields and options:

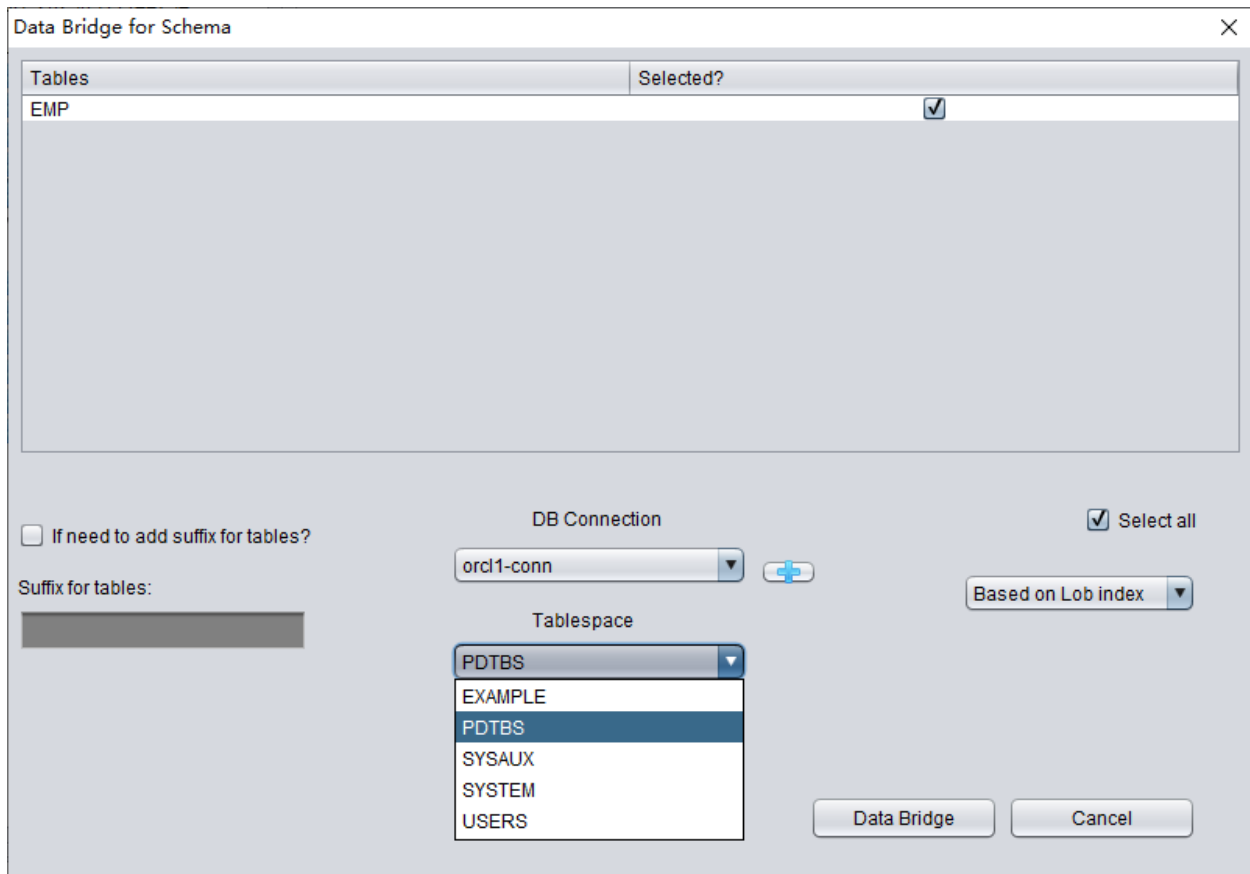
- Connection Name: orcl1-conn
- Username: pd
- Password: *****
- Hostname: localhost
- Port: 1521
- Radio buttons: SID, Service name
- Service name: ORCL1
- Buttons: Save, Test (highlighted with a red rectangle), Cancel

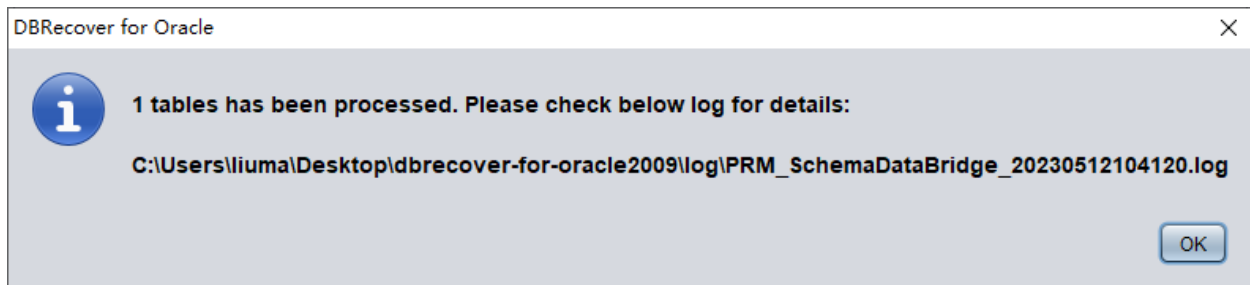
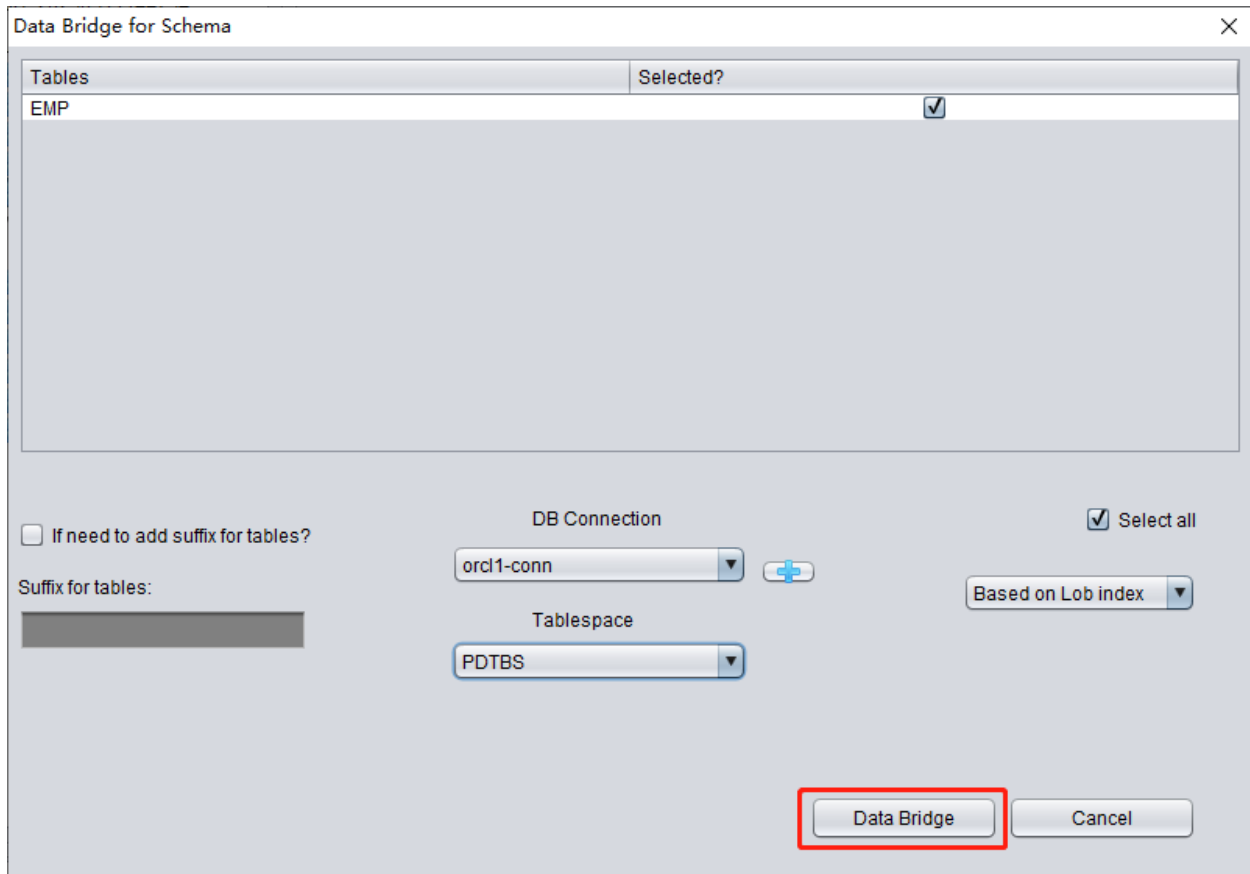


Si réussi, cliquez sur SAVE pour sauvegarder :









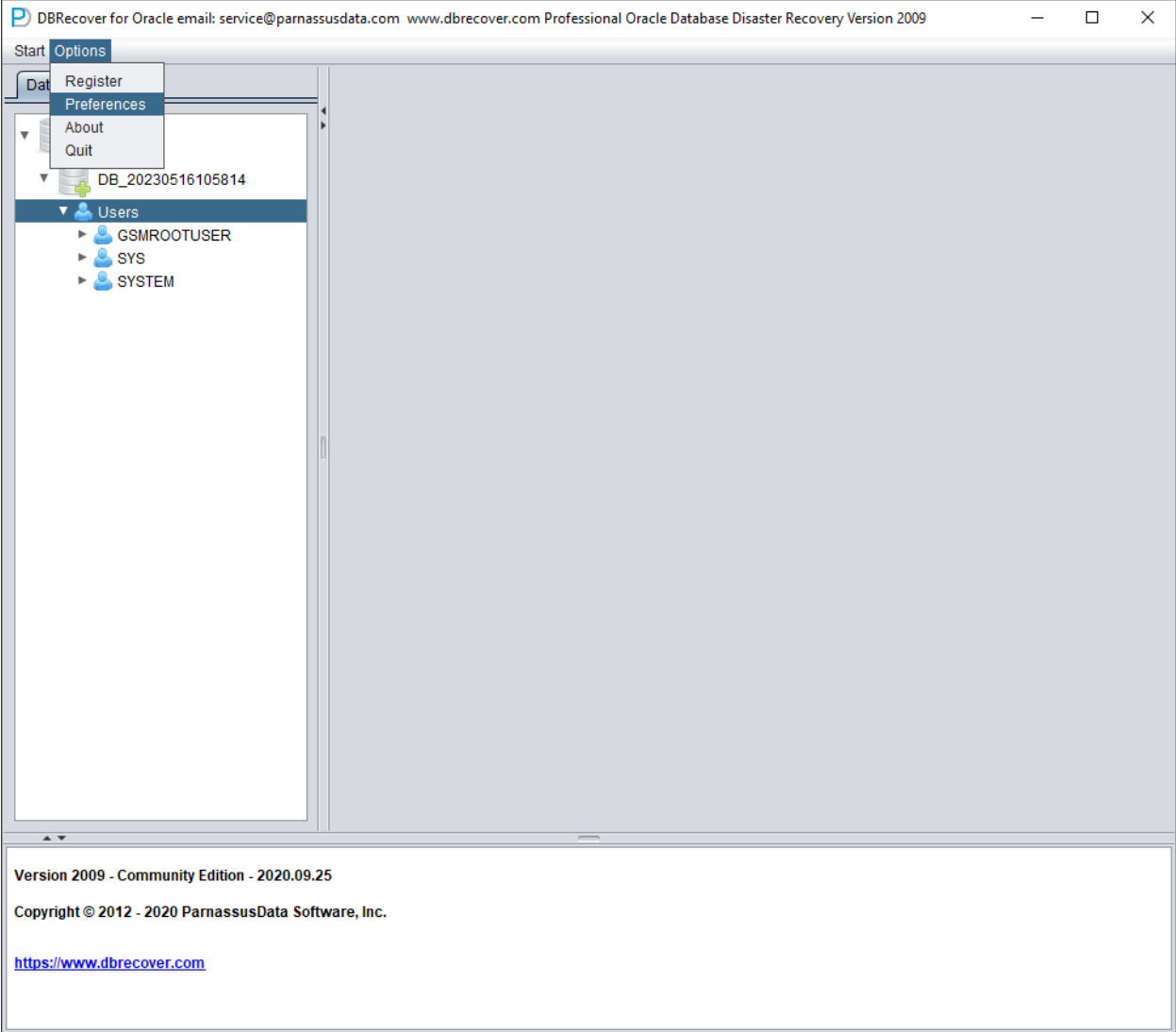
Vérifiez les données :

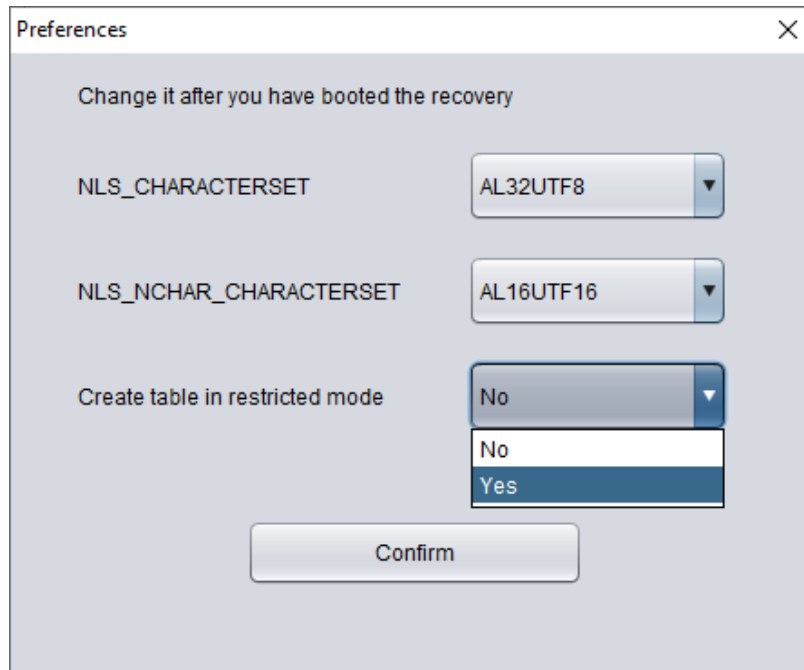
```
SQL> show parameter db_name
NAME TYPE VALUE
-----
db_name string ORCL1
SQL> select count(*) from pd.emp;
COUNT(*)
```


Introduction au mode WIDE TABLE

Remarque : le transfert de données ci-dessus utilise par défaut le mode table large (wide table mode) pour transférer les données, c'est-à-dire qu'il convertit par défaut tous les types de champs CHAR, NCHAR, VARCHAR, NVARCHAR en leur longueur maximale, c'est-à-dire 2000 ou 4000. Cela est fait pour éviter le problème potentiel d'insertion de chaînes de récupération en raison de champs trop courts.

Si vous ne souhaitez pas utiliser le mode table large, vous pouvez cliquer sur le menu Options => Préférences





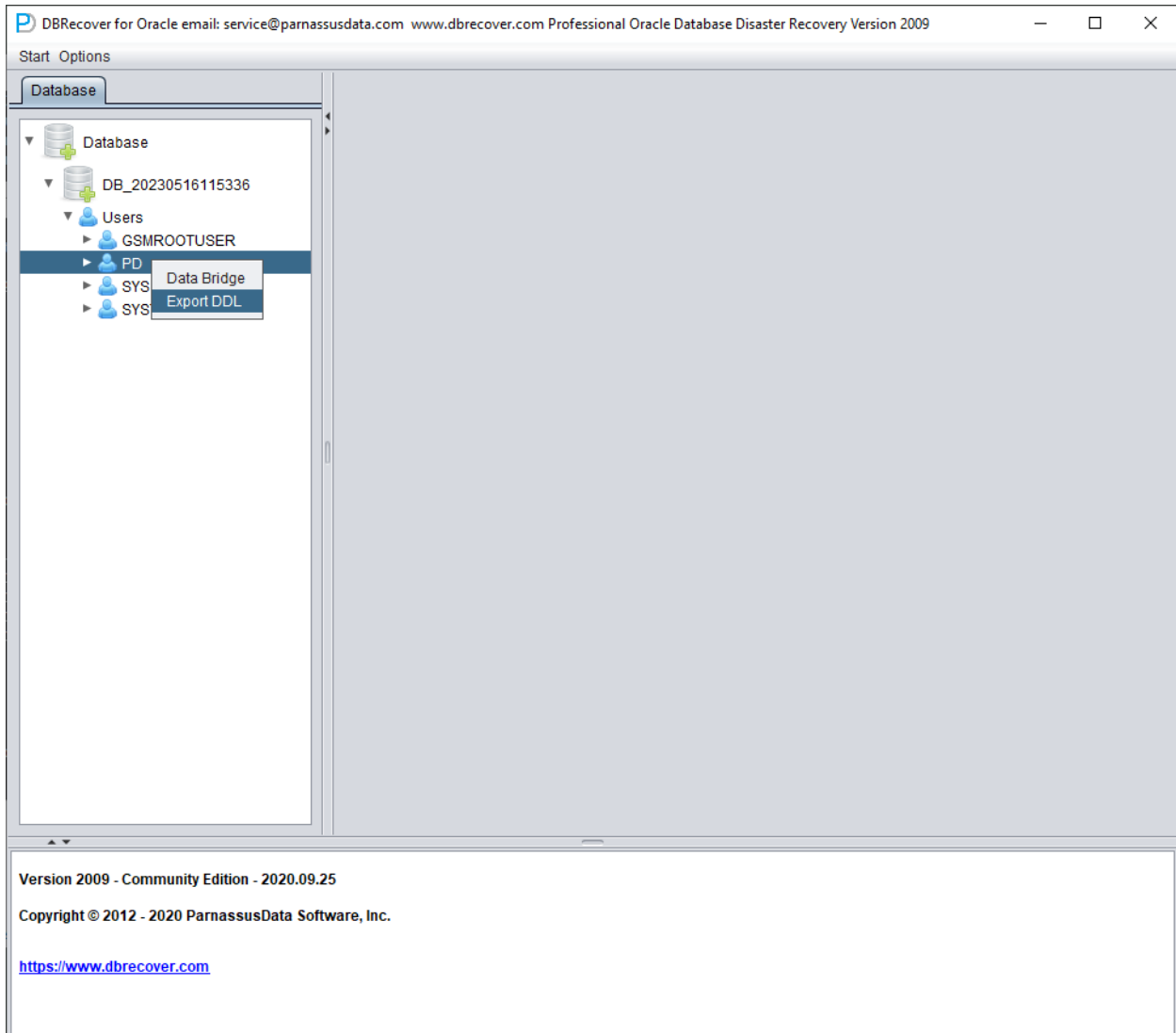
Ci-dessus, en cochant "Yes" dans le menu déroulant "Create table in restricted mode", vous n'utiliserez plus le mode de création de table large.

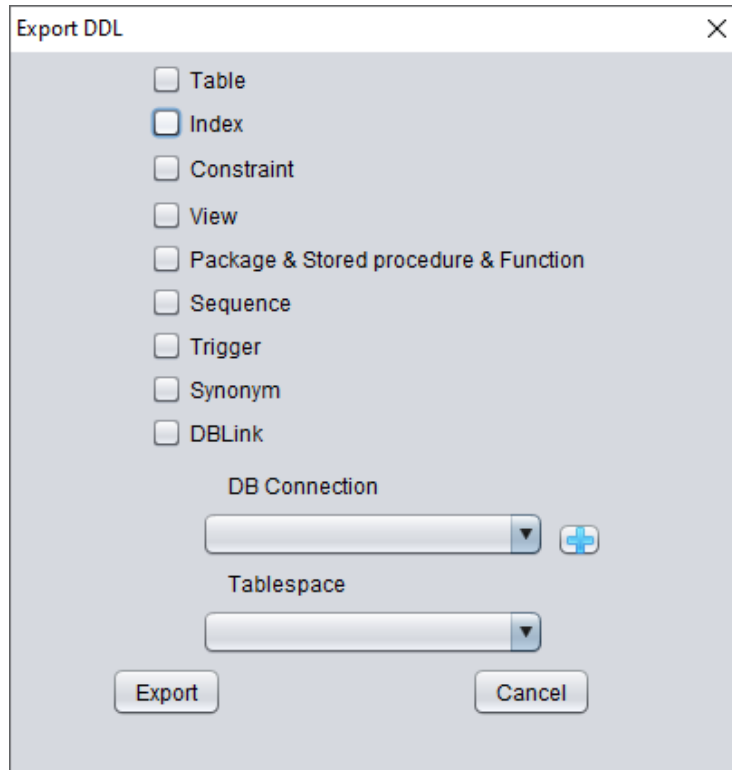
Introduction à la fonction EXPORT DDL

Ci-dessus, nous avons effectué une opération de récupération sur une seule table de données SCHEMA, dont les objets de récupération comprennent : création de la table de données correspondante, insertion des données récupérables.

Pour la récupération d'objets tels que les index, les contraintes, les vues, les déclencheurs, etc., vous pouvez obtenir la fonction EXPORT DDL.

Sélectionnez le SCHEMA à récupérer, faites un clic droit et sélectionnez la fonction EXPORT DDL :

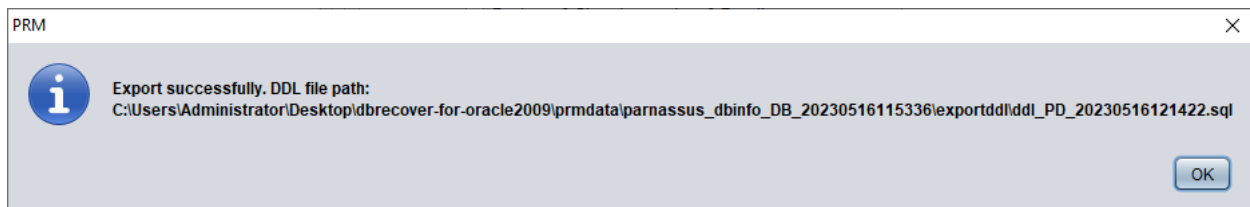
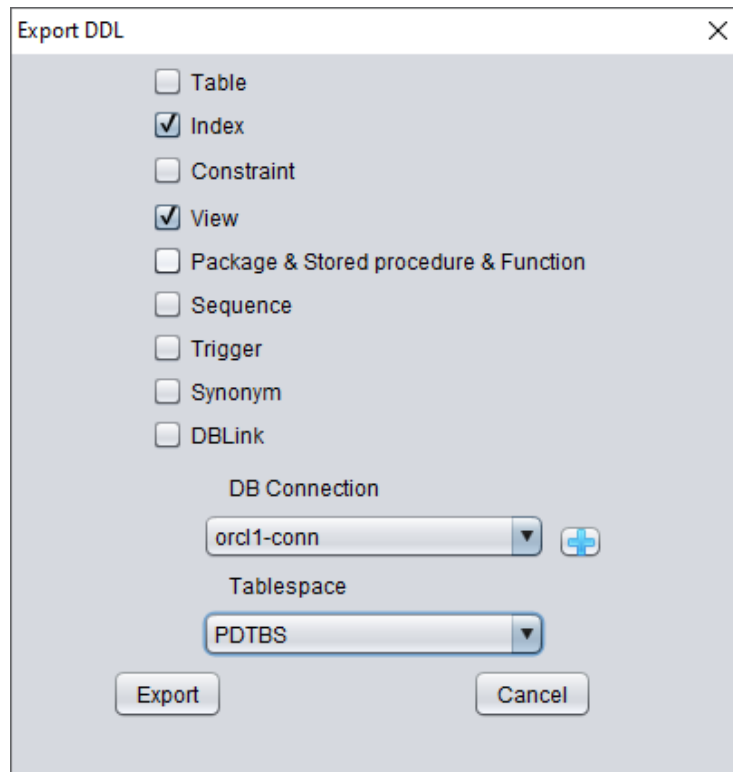




Les types d'objets récupérables comprennent :

- Instruction de création de table create table statement (notez que cela n'inclut pas les informations de partition, etc.)
- Instruction de création d'index create index statement (notez que cela n'inclut pas les informations de partition, etc.)
- Contrainte constraint
- Vue view
- Package de procédures stockées et fonctions Package & Stored Procedure & Function
- Séquence sequence
- Déclencheur trigger
- Synonyme synonym
- Lien de base de données DBlink

Ici aussi, choisissez les informations de connexion de la base de données que vous avez saisies précédemment pour traiter temporairement les informations DDL.



Une fenêtre pop-up indique le chemin du fichier SQL DDL, regardez ce fichier :

```
1 EXPORT DDL will not work until you have a valid license key! contact service@parnassusdata.com
2 DDL导出功能需要在购买使用许可证后使用, 购买联系 13764045638
3 Primary key constraints will be included in this file , but other type constraints pls check pd_constraint_details in target
4 主键约束信息可以通过exportddl直接获得, 其他类型约束请参考目标数据库中的pd_constraint_details视图
5 if you have old exp or expdp backup , you should import the metadata from dmp file (imp rows=no for exp , impdp content=metadat
6 若你有老的exp或expdp的dmp文件, 则建议从dmp文件中导入元数据信息 (imp使用rows=no, impdp使用content=metadata_only可仅导入结构信息), expc
7
8 alter session set current_schema=PD;
9 ----- Start usertablespace DDL -----
10
11
12 create USER ANONYMOUS IDENTIFIED BY VALUES ''';
13 create USER SYS IDENTIFIED BY VALUES 'S:29373CC71D0D61C29A4CAE771BC989D92DFD1F673F4205C37FDFCD094D1;T:5B1133CACDB2336BE5F9E75
14 create USER AUDSYS IDENTIFIED BY VALUES 'S:00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
15 create USER SYSTEM IDENTIFIED BY VALUES 'S:E3EE494A5DF42DC11055F9F17CEE299BF6C789A59576EEC6BB623C352BA2;T:02AAEFCE2F572643493
16 create USER SYSBACKUP IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
17 create USER SYSDBG IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
18 create USER SYSRM IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
19 create USER SYSRAC IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
20 create USER OUTLN IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
21 create USER XS$NULL IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000;
22 create USER GSMADMIN_INTERNAL IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
23 create USER GSMUSER IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
24 create USER GSMROOTUSER IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
25 create USER DIP IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
26 create USER DBSFUSER IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
27 create USER ORACLE_OCM IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
28 create USER SYS$UMF IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
29 create USER DBSNMP IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
30 create USER APPQOSSYS IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
31 create USER GSMCATUSER IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
32 create USER GGSYS IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
33 create USER XDB IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
34 create USER WMSYS IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
35 create USER MDDATA IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
36 create USER OJVMSYS IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
37 create USER CTXSYS IDENTIFIED BY VALUES 'S:C8FF0ADB3343B713BFD77466E33F4F615893318668FAB9B3BB9C68CC201;T:93DABE43D29FDE41512
38 create USER ORDSYS IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
39 create USER ORDDATA IDENTIFIED BY VALUES 'S:0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

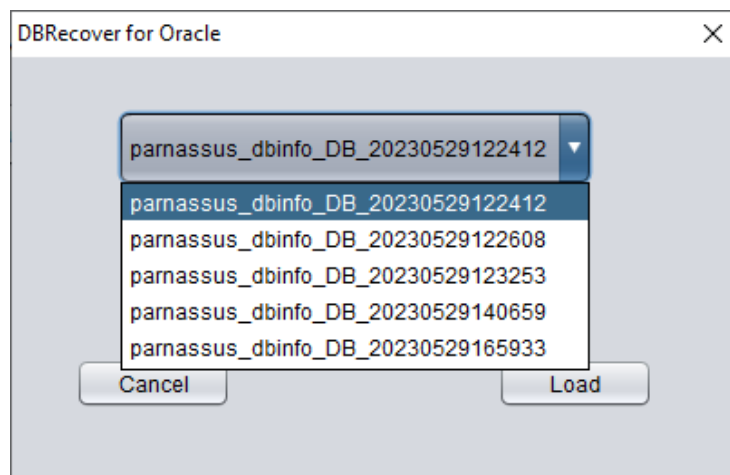
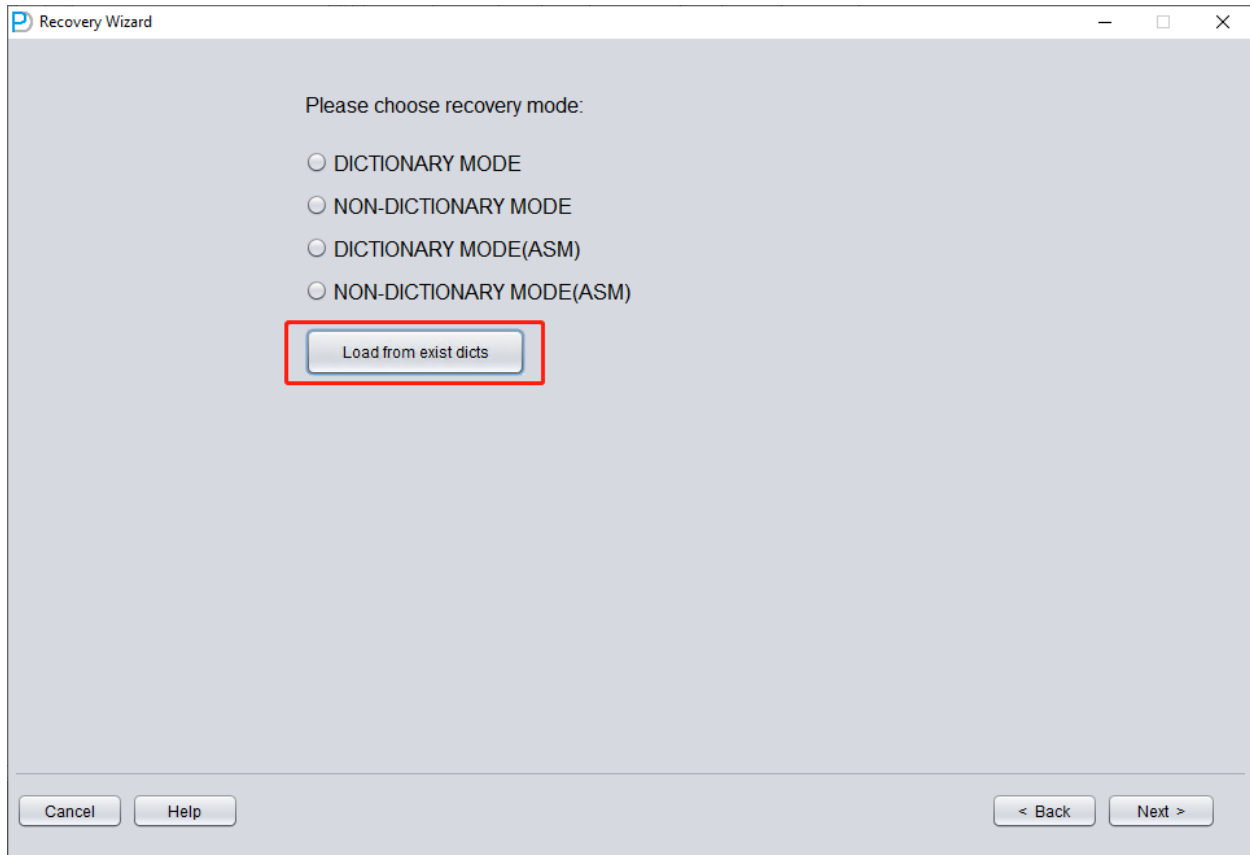
Remarque : la fonction EXPORTDDL ne peut être utilisée normalement qu'après l'enregistrement d'une licence d'entreprise valide (LICENSE KEY) !

Les instructions de création d'index, de vues, etc., dans le fichier SQL DDL ci-dessus doivent être copiées et exécutées par l'utilisateur sous l'utilisateur correspondant de la base de données.

Si l'utilisateur dispose d'anciens fichiers dmp exp ou expdp, il est recommandé d'importer les informations de métadonnées à partir des fichiers dmp (imp utilise rows=no, impdp utilise content=metadata_only pour importer uniquement les informations de structure), la fonction exportddl manque de certaines informations de métadonnées, comme l'autorisation d'objets et les clés étrangères, etc.

Introduction à la fonction LOAD FROM EXIST DICTS :

En cas de non-réponse, de blocage, d'erreur, etc., lors du processus de récupération réel, vous pouvez utiliser la fonction LOAD FROM EXIST DICTS pour charger directement l'état de récupération précédent après le redémarrage de DBRECOVER.

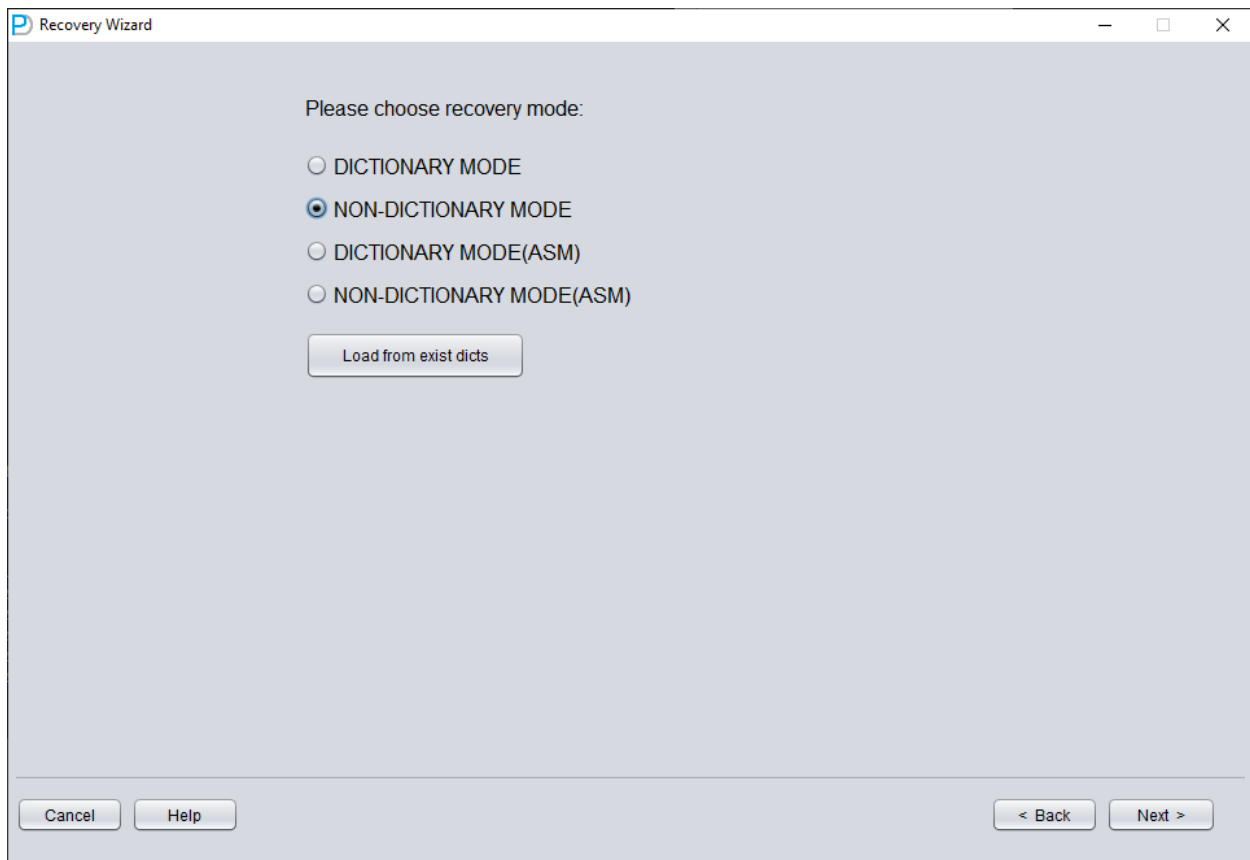


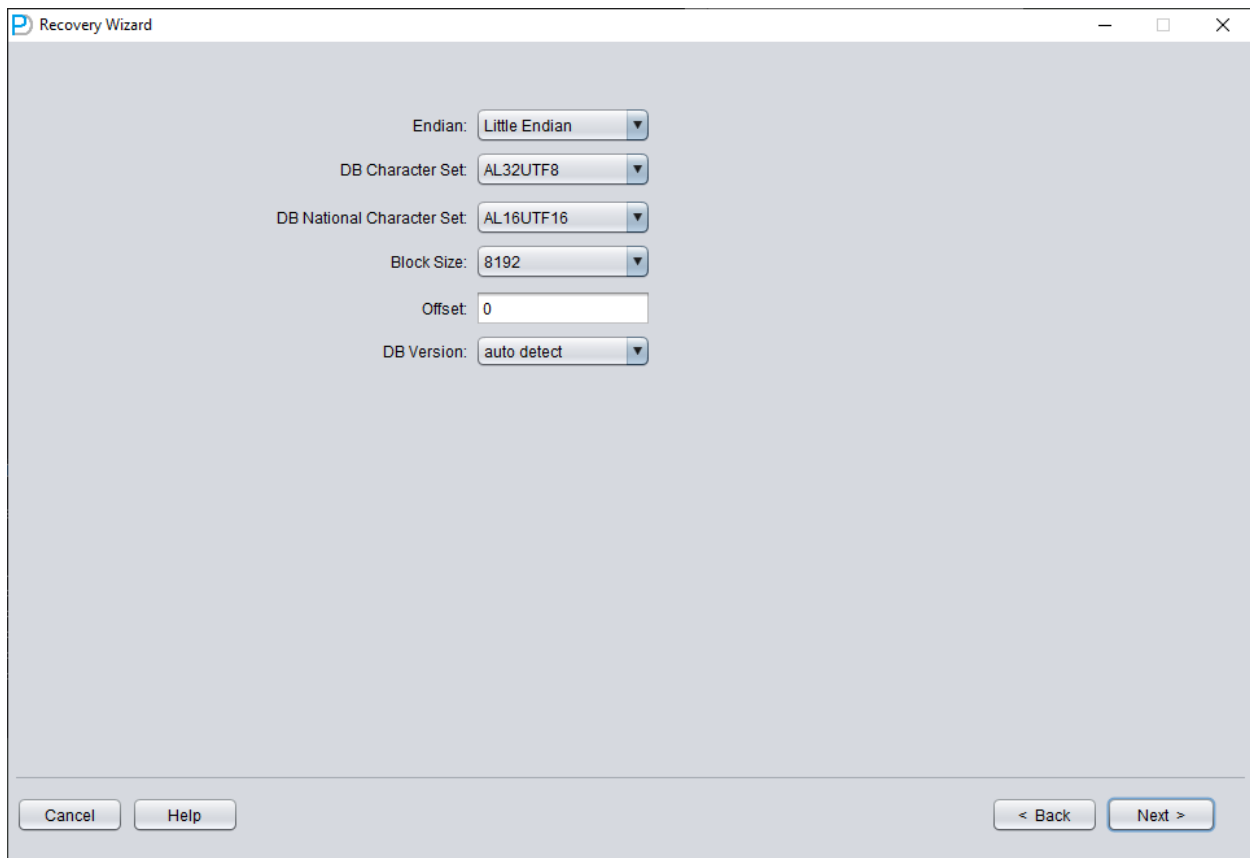
L'état de récupération est trié par ordre chronologique, choisissez convenablement et cliquez sur le bouton LOAD pour charger. Le mode dictionnaire (DICTIONARY-MODE) et le mode non-dictionnaire (NON-DICTIONARY MODE) peuvent tous deux utiliser cette fonction de chargement rapide pour éviter les opérations répétitives.

Scénario de récupération 2 Suppression erronée ou perte totale de l'espace de table SYSTEM

L'administrateur du système SA de la société D a accidentellement supprimé le fichier de données où se trouve l'espace de table SYSTEM d'une base de données, ce qui a rendu la base de données complètement inopérable et les données inaccessibles. En l'absence de sauvegarde, vous pouvez utiliser DBRECOVER pour récupérer les données.

Dans ce scénario, après avoir lancé DBRECOVER, entrez dans le Recovery Wizard et choisissez le mode "Non-Dictionary mode" (mode non-dictionnaire) :

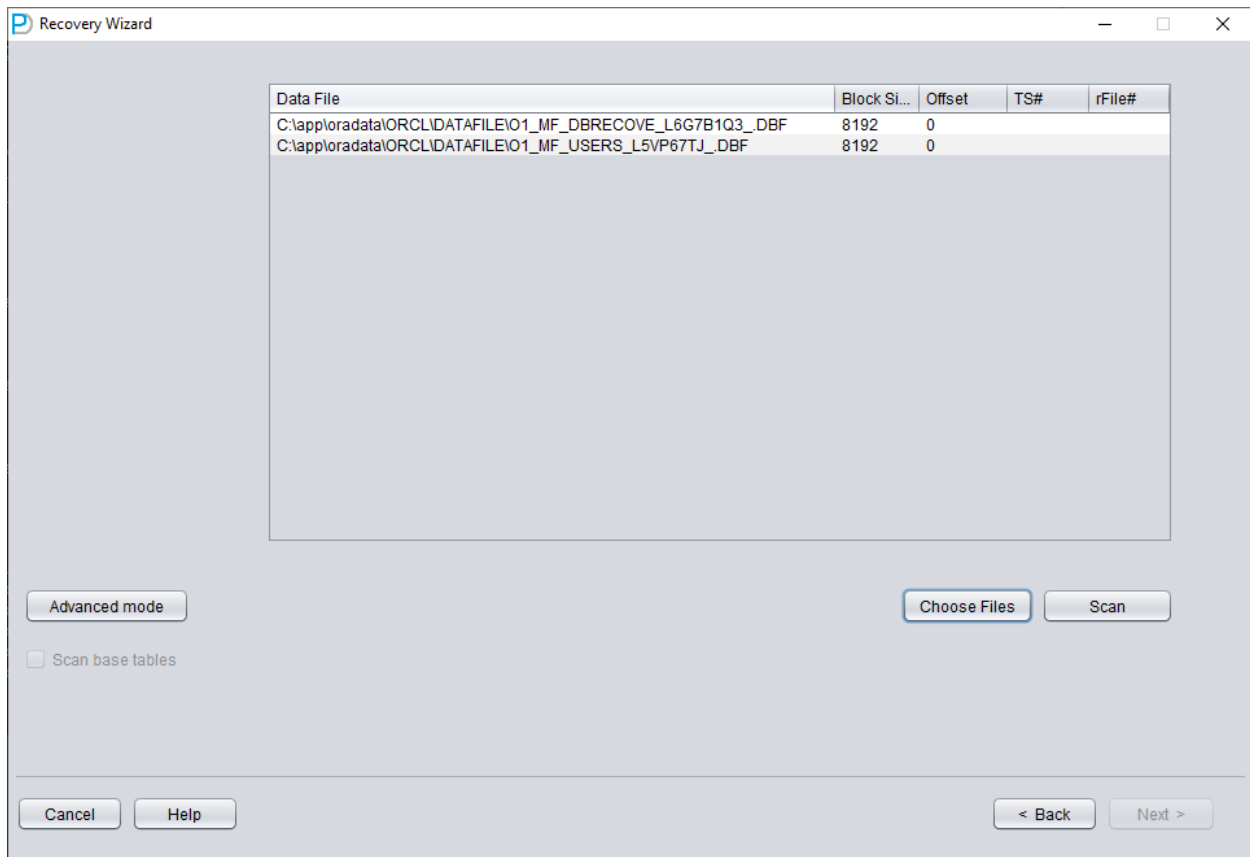




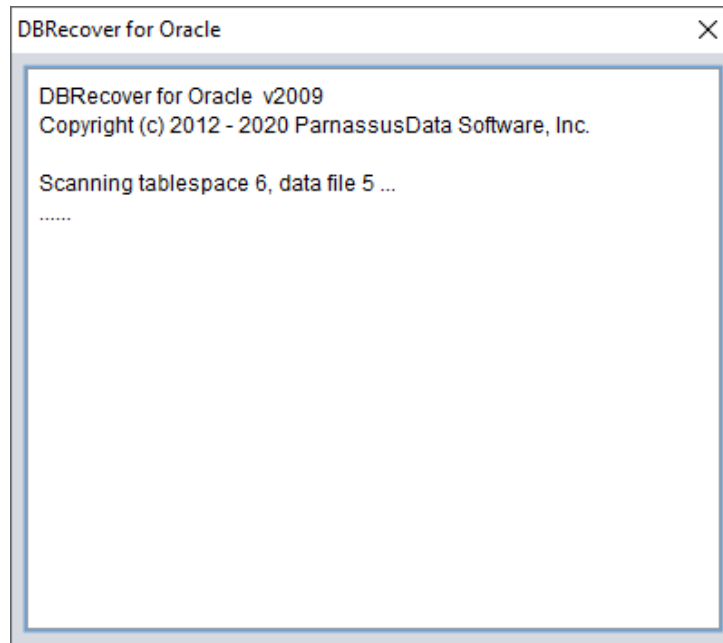
Par la suite, il faut choisir le bon jeu de caractères, sinon les données suivantes apparaîtront en désordre.

Dans le mode NoN-dictionary, l'utilisateur doit spécifier le jeu de caractères et le jeu de caractères national, car après la perte de l'espace de table SYSTEM, les informations sur le jeu de caractères de la base de données ne peuvent pas être obtenues normalement, donc l'entrée de l'utilisateur est nécessaire. Seuls les bons paramètres de jeu de caractères et les packages de langue nécessaires peuvent garantir l'extraction normale de plusieurs langues en mode No-Dictionary.

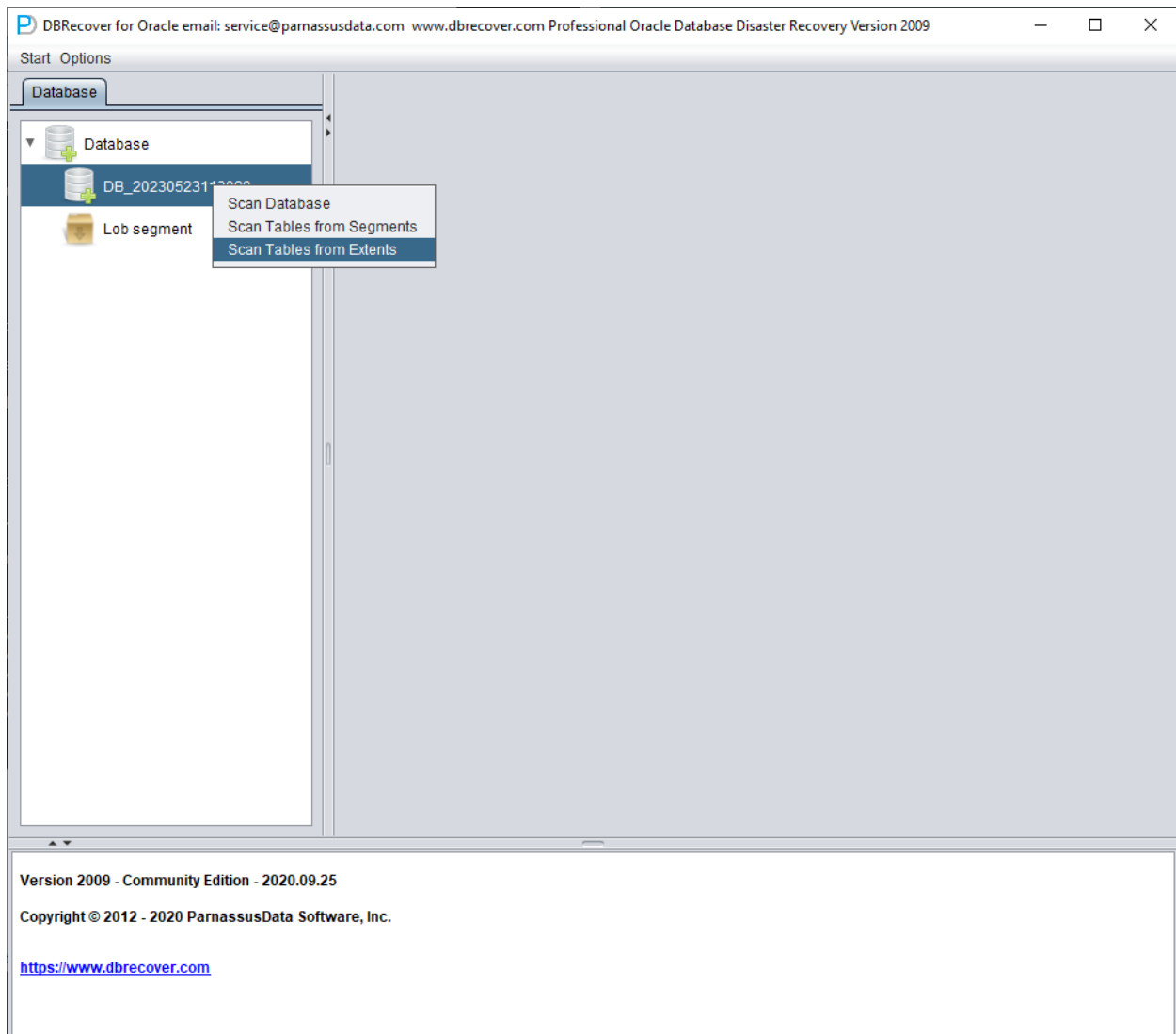
Comme dans la démonstration du scénario 1, entrez tous les fichiers de données disponibles pour l'utilisateur (à l'exception des fichiers temporaires) et définissez correctement le Block Size et l'OFFSET :



Cliquez ensuite sur SCAN, la fonction SCAN est de scanner toutes les informations de données sur tous les fichiers de données.



Ensuite, faites un clic droit sur le nœud de la base de données dans l'arborescence de gauche et choisissez SCAN EXTENT. Utilisez le mode SCAN TABLE FROM SEGMENTS uniquement lorsque vous pouvez confirmer que tous les fichiers de données (à l'exception de SYSTEM01.DBF) sont disponibles. L'avantage de ce mode est que sa vitesse est légèrement plus rapide, mais en cas de fichiers de données incomplets ou endommagés, son degré de récupération est inférieur à celui du mode SCAN EXTENT.



Après avoir terminé Scan Tables From Extents, vous pouvez ouvrir l'arborescence sur le côté gauche de l'interface principale :

DBRecover for Oracle email: service@parnassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

Database

Database

- DB_20230523113028
 - Extents
 - obj63328
 - obj63330
 - obj63332
 - obj63336
 - obj63338
 - obj63341
 - obj63352
 - obj63378
 - obj63380
 - obj63388
 - obj63389
 - obj63391
 - obj74036
 - obj74038
 - obj74041
 - obj74042
 - obj74046
 - Lob segment

obj74046 :

Col#	Seen Count	Max Size	NULL PCT	String	NString	Number	Date	Timestamp	Timestamp Zone	Clob	Blot
1	1500	3	0	0	0	1500	0	0	0	0	0
2	1500	6	0	1500	644	0	0	0	0	0	0
3	1500	9	0	1500	428	0	0	0	0	0	0
4	1500	3	7	0	0	1393	0	0	0	0	0

Sample data analysis:

col1	col2	col3	col4	col5	col6	col7	col8
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30

Try to analyze UNKNOWN column type:

Columns	Date	Number	String(VARCHAR2 CHAR)	Timestamp	Timestamp with time zone	NString(NVARCHAR2 NC)

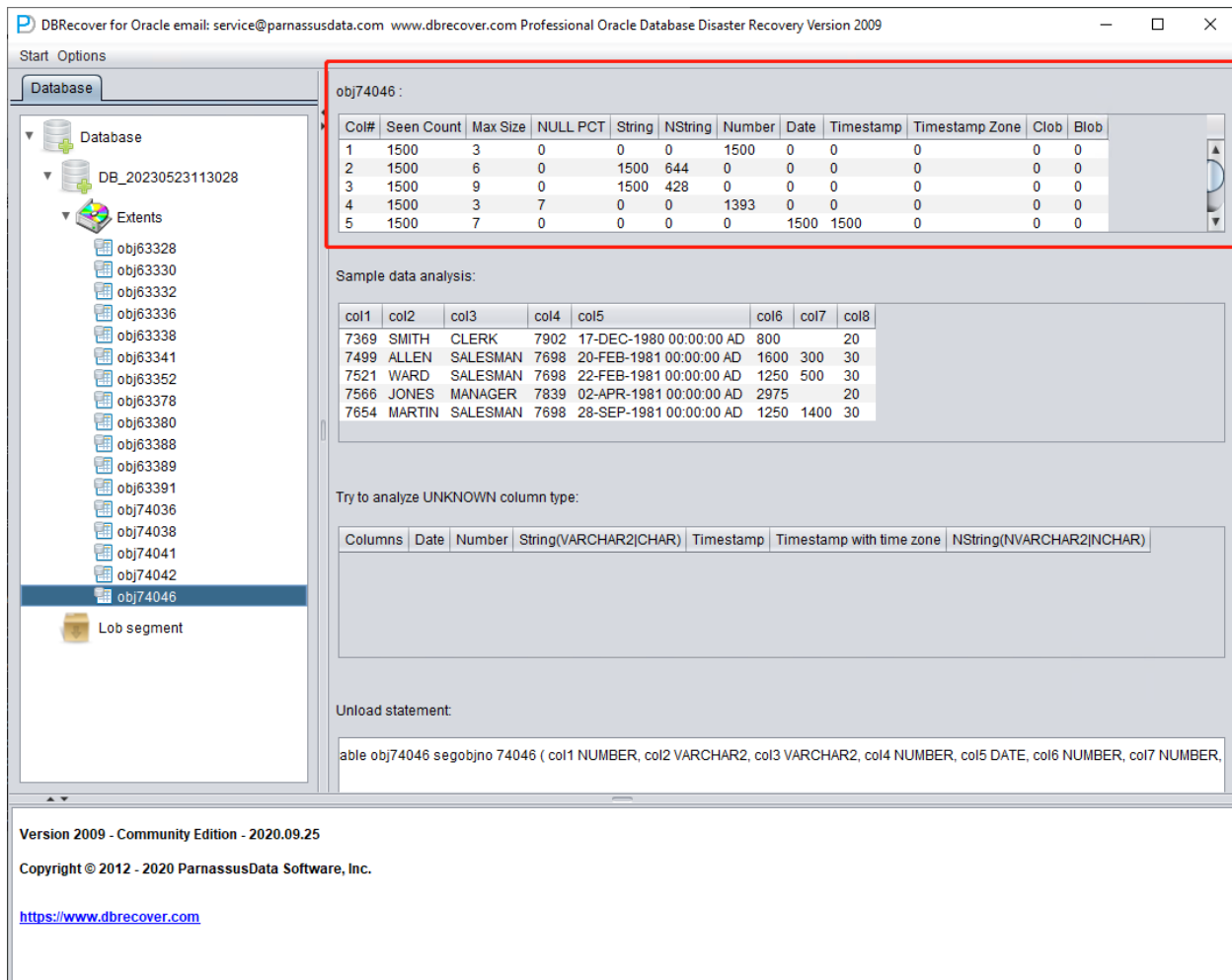
Unload statement:

```
col1 NUMBER, col2 VARCHAR2, col3 VARCHAR2, col4 NUMBER, col5 DATE, col6 NUMBER, col7 NUMBER, col8 NUMBER )
```

Version 2009 - Community Edition - 2020.09.25
 Copyright © 2012 - 2020 ParnassusData Software, Inc.
<https://www.dbrecover.com>

Chaque nœud de l'arborescence représente un fragment de données d'une table heap ordinaire ou d'une partition, son nom est obj+ DATA OBJECT ID enregistré sur le segment de données.

Sélectionnez un nœud et observez la barre latérale de l'interface principale :



Analyse des types de champs

En raison de la perte de l'espace de table SYSTEM, le mode Non-Dictionary manque d'informations structurales sur la table de données, ces informations comprennent le nom et le type de champ de la table, et dans ORACLE, ces informations sont toutes enregistrées comme des informations de dictionnaire, elles ne sont pas stockées sur la table de données. Lorsqu'un utilisateur n'a que l'espace de table où se trouvent les données de l'application, il doit deviner le type de chaque champ en se basant sur les données ROW du segment de données. Ici, nous pouvons analyser jusqu'à une dizaine de types de données courants :

- String chaînes de caractères : inclut char, varchar
- NString chaînes de caractères nationales : nchar, nvarchar
- Number type numérique

- Date type date
- TimeStamp type horodatage
- TimeStamp Zone type horodatage avec zone
- CLOB
- BLOB

Analyse des données d'échantillon Sample Data Analysis :

DBRecover for Oracle email: service@parnassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

Database

obj74046 :

Col#	Seen Count	Max Size	NULL PCT	String	NString	Number	Date	Timestamp	Timestamp Zone	Clob	Blob
1	1500	3	0	0	0	1500	0	0	0	0	0
2	1500	6	0	1500	644	0	0	0	0	0	0
3	1500	9	0	1500	428	0	0	0	0	0	0
4	1500	3	7	0	0	1393	0	0	0	0	0
5	1500	7	0	0	0	0	1500	1500	0	0	0

Sample data analysis:

col1	col2	col3	col4	col5	col6	col7	col8
7369	SMITH	CLERK	7902	17-DEC-1980 00:00:00 AD	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981 00:00:00 AD	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981 00:00:00 AD	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981 00:00:00 AD	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981 00:00:00 AD	1250	1400	30

Try to analyze UNKNOWN column type:

Columns Date Number String(VARCHAR2[CHAR]) Timestamp Timestamp with time zone NString(NVARCHAR2[INCHAR])

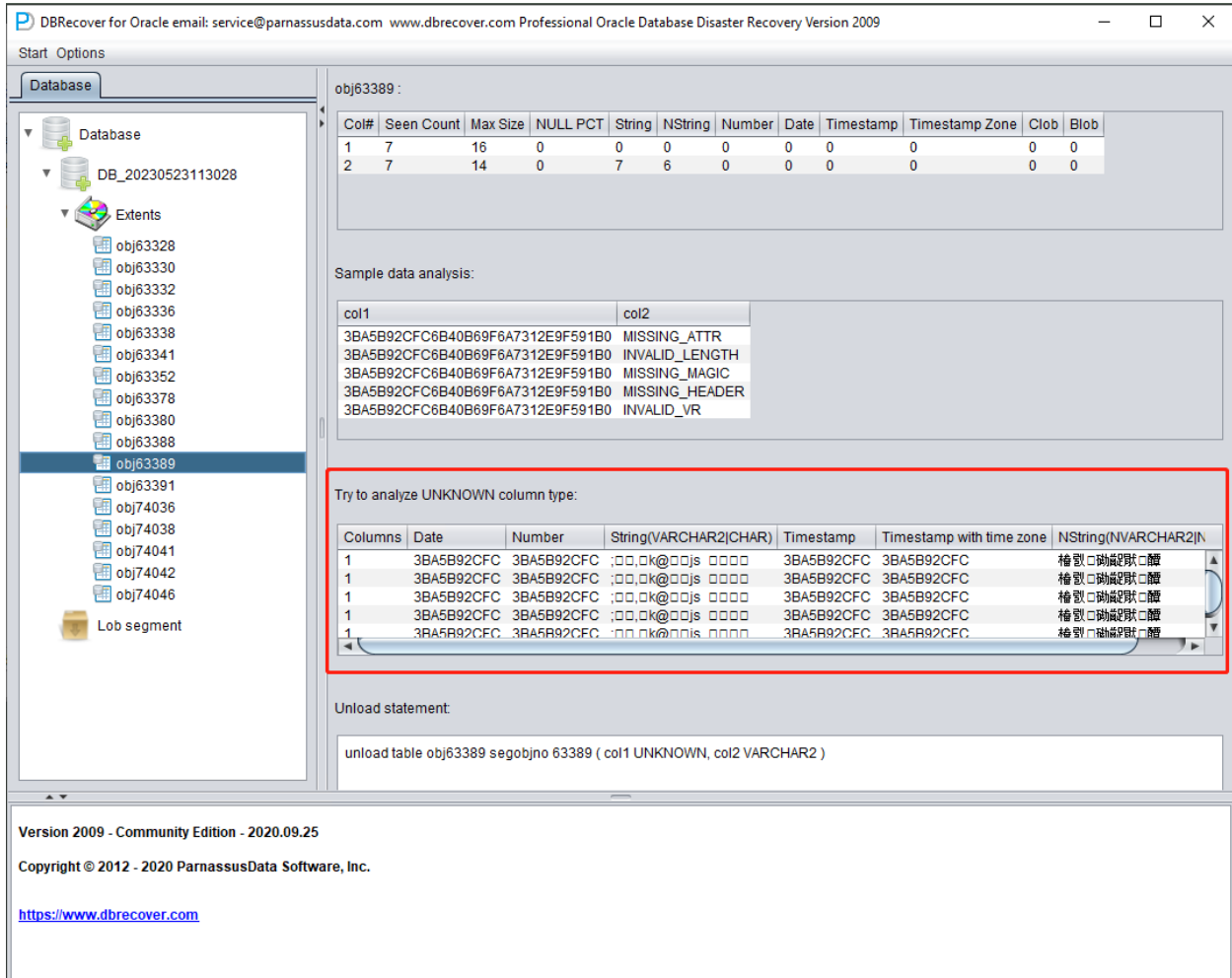
Unload statement

able obj74046 segobjno 74046 (col1 NUMBER, col2 VARCHAR2, col3 VARCHAR2, col4 NUMBER, col5 DATE, col6 NUMBER, col7 NUMBER,

Version 2009 - Community Edition - 2020.09.25
 Copyright © 2012 - 2020 ParnassusData Software, Inc.
<https://www.dbrecover.com>

Cette section analyse 10 données en fonction des résultats de l'analyse des types de champs et affiche les résultats de l'analyse. Les données d'échantillon peuvent aider l'utilisateur à

comprendre l'état réel des données stockées dans ce segment de données. Si le nombre d'enregistrements sur le segment de données est inférieur à 10, tous les enregistrements sont affichés.



TRY TO ANALYZE UNKNOWN column type:

ESSAYER D'ANALYSER le type de colonne inconnu :

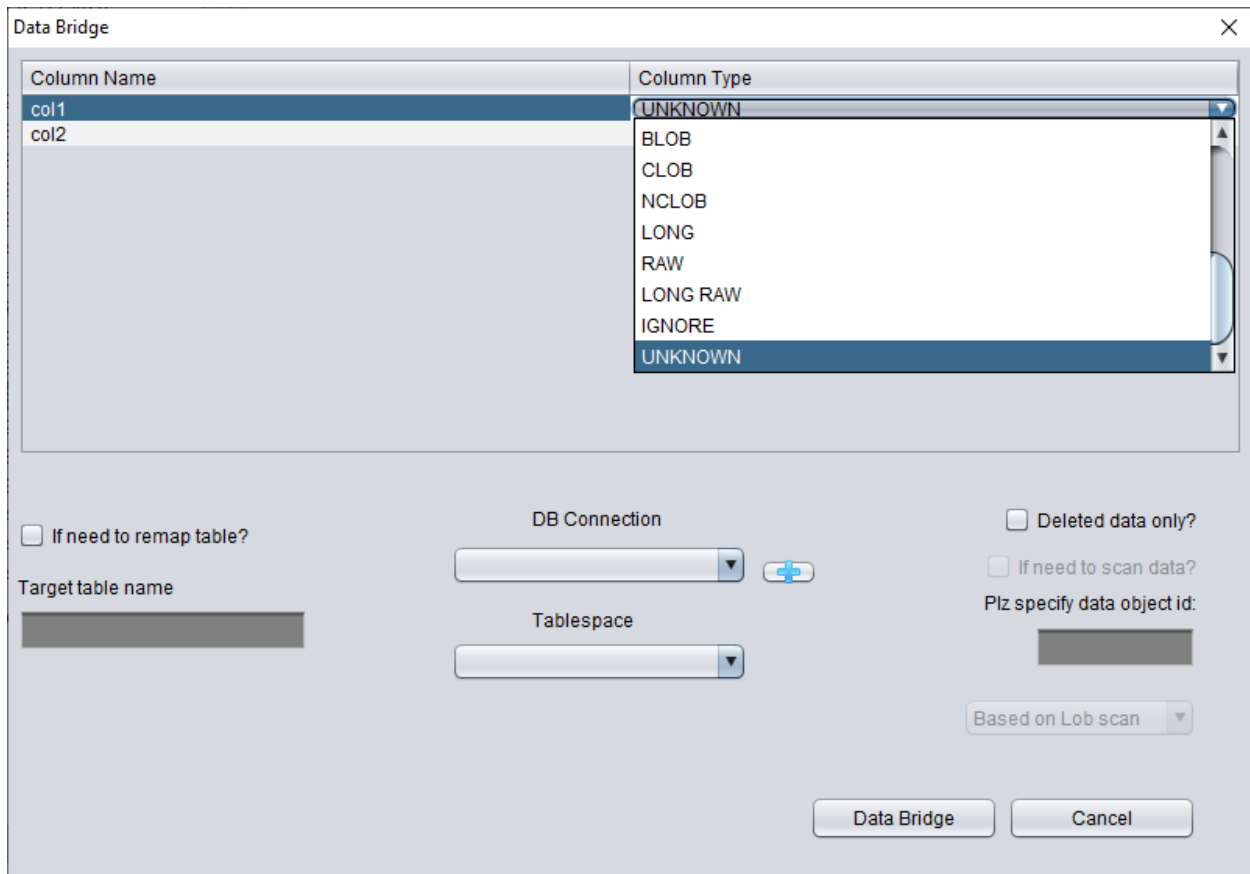
Cette section concerne les champs dont le type ne peut pas être pleinement confirmé par la fonction d'analyse des champs, en essayant d'analyser avec divers types de champs et de les présenter à l'utilisateur, afin que l'utilisateur puisse juger de quel type il s'agit.

Les champs dont le type ne peut pas être confirmé sont généralement dans les situations suivantes :

1. RAW ou LONG RAW
2. Types de données non pris en charge, y compris : XDB.XDB\$RAW_LIST_T, XMLTYPE, types définis par l'utilisateur, etc.
3. Le bloc de données lui-même est gravement endommagé

Dans ce mode "Non-Dictionary Mode" (mode non-dictionnaire), vous pouvez également utiliser le mode conventionnel et le mode de transfert de données. Par rapport au mode dictionnaire, la principale différence est qu'en mode non-dictionnaire, l'utilisateur peut décider du type de champ lors du transfert de données, comme le montre la figure ci-dessous, où certains types de champs sont UNKNOWN (inconnu).

Si l'utilisateur connaît la structure de cette table lors de sa conception (qui peut également provenir de la documentation du développeur de l'application), il peut alors choisir lui-même le bon type de colonne (Column Type) afin de pouvoir transférer les données de cette table à la base de données cible en douceur.



Scénario de récupération 3 : Cas de fichiers de données endommagés par un logiciel malveillant de rançon

Les logiciels malveillants de rançon (ransomware malware) peuvent crypter tout ou partie du contenu des fichiers de données ORACLE. Comme la taille des fichiers de données d'ORACLE est généralement importante, le cryptage complet peut prendre beaucoup de temps, donc certains logiciels malveillants de rançon peuvent choisir de crypter uniquement l'espace continu ou aléatoire de l'en-tête du fichier de données ORACLE.

Pour ce type de cryptage partiel, nous pouvons essayer d'utiliser DBRECOVER pour récupérer les données.

Comme l'en-tête du fichier de données est endommagé, nous devons observer le contenu de SYSTEM01.DBF pour comprendre les informations telles que le numéro de l'espace de table (TS#) et le numéro de fichier relatif (RFILE#) de chaque fichier de données.

Voici la liste des fichiers de données :

```
Administrator: Command Prompt
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

C:\Users\Administrator>cd C:\Users\Administrator\Desktop\DATAFILE

C:\Users\Administrator\Desktop\DATAFILE>dir
Volume in drive C is System Drive
Volume Serial Number is 5EB5-5EB4

Directory of C:\Users\Administrator\Desktop\DATAFILE

05/29/2023  11:35 AM  <DIR>          .
05/29/2023  11:22 AM  <DIR>          ..
05/29/2023  11:22 AM             524,296,192  01_MF_APP01_L782YY4Y_.DBF.eking
05/29/2023  11:22 AM             104,865,792  01_MF_APP01_L782ZBM3_.DBF.eking
05/29/2023  11:22 AM             104,865,792  01_MF_APP01_L782ZCP1_.DBF.eking
05/29/2023  11:22 AM             524,296,192  01_MF_APP02_L782Z07W_.DBF.eking
05/29/2023  11:22 AM             104,865,792  01_MF_APP02_L7830DTG_.DBF.eking
05/29/2023  11:22 AM             104,865,792  01_MF_APP02_L7830FJ6_.DBF.eking
05/29/2023  11:22 AM             524,296,192  01_MF_DBRECOVE_L6G7B1Q3_.DBF.eking
05/29/2023  11:22 AM             1,069,555,712  01_MF_SYSAUX_L5VP5QJ8_.DBF.eking
05/29/2023  11:22 AM             964,698,112  01_MF_SYSTEM_L5VP4N7Y_.DBF.eking
05/29/2023  07:03 AM             135,274,496  01_MF_TEMP_L5VPCQGO_.TMP.eking
05/29/2023  11:22 AM              68,165,632  01_MF_UNDOTBS1_L5VP66PM_.DBF.eking
05/29/2023  11:22 AM              10,493,952  01_MF_USERS_L5VP67TJ_.DBF.eking
          12 File(s)  4,240,539,648 bytes
          2 Dir(s)  6,546,952,192 bytes free

C:\Users\Administrator\Desktop\DATAFILE>
```

- 01_MF_APP01_L782YY4Y_.DBF.eking
- 01_MF_APP01_L782ZBM3_.DBF.eking
- 01_MF_APP01_L782ZCP1_.DBF.eking
- 01_MF_APP02_L782Z07W_.DBF.eking
- 01_MF_APP02_L7830DTG_.DBF.eking
- 01_MF_APP02_L7830FJ6_.DBF.eking
- 01_MF_DBRECOVE_L6G7B1Q3_.DBF.eking
- 01_MF_SYSAUX_L5VP5QJ8_.DBF.eking
- 01_MF_SYSTEM_L5VP4N7Y_.DBF.eking
- 01_MF_TEMP_L5VPCQGO_.TMP.eking

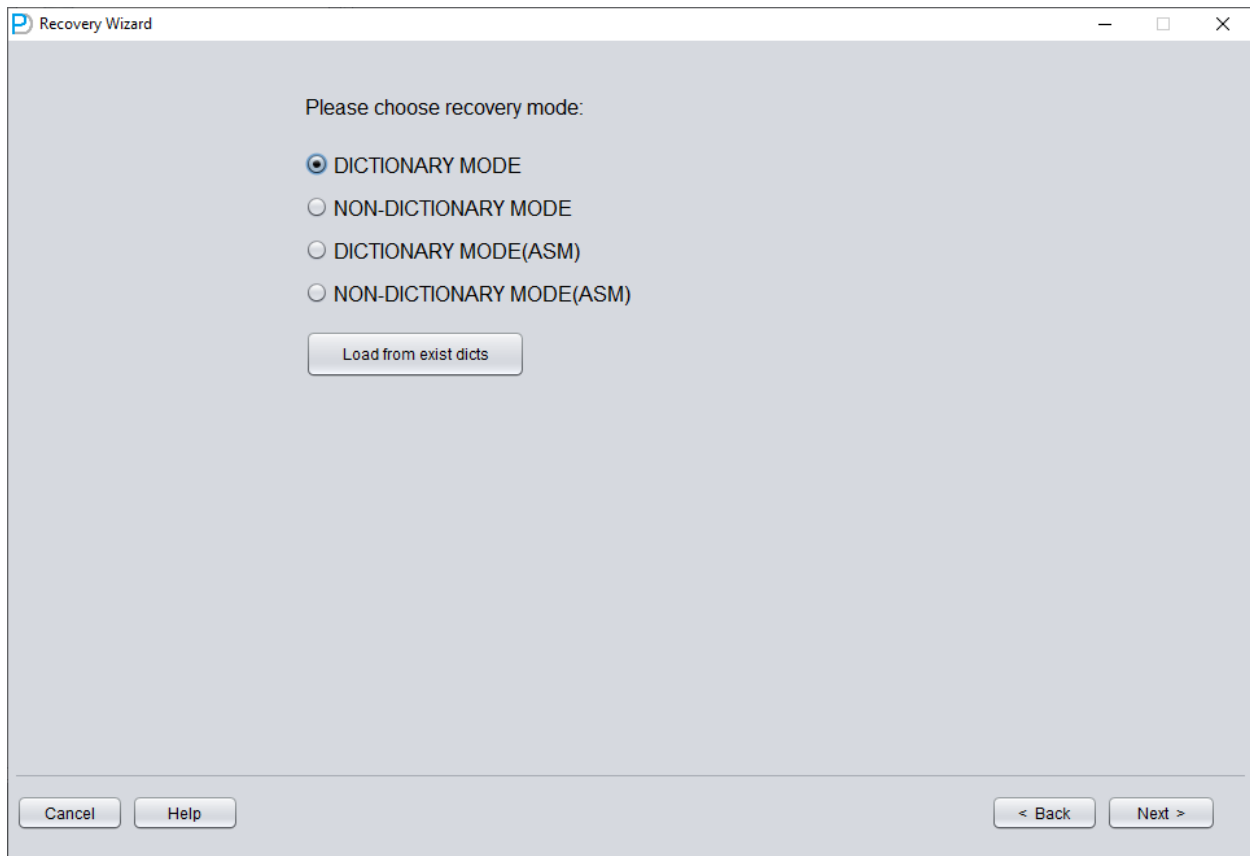
01_MF_UNDOTBS1_L5VP66PM_.DBF.eking

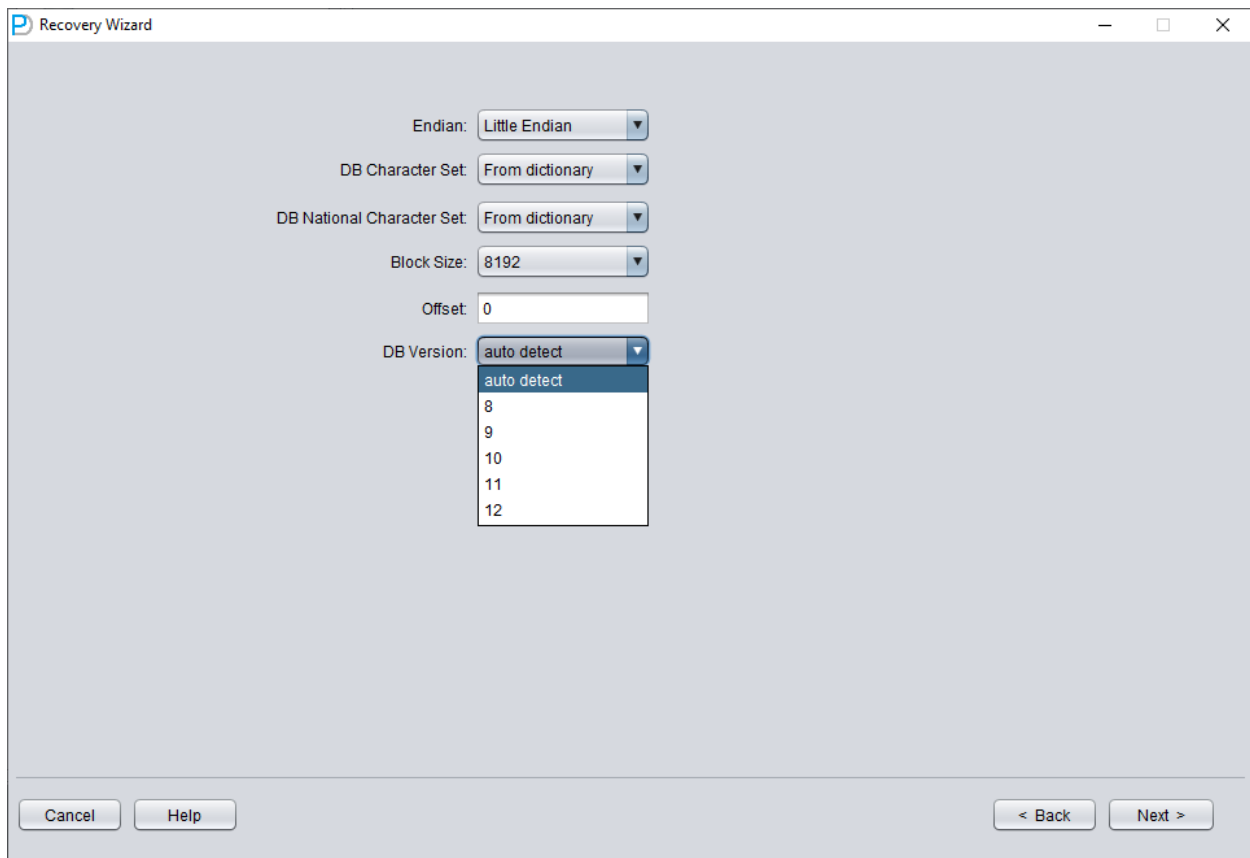
01_MF_USERS_L5VP67TJ_.DBF.eking

L'exemple ci-dessus est crypté avec le suffixe eking

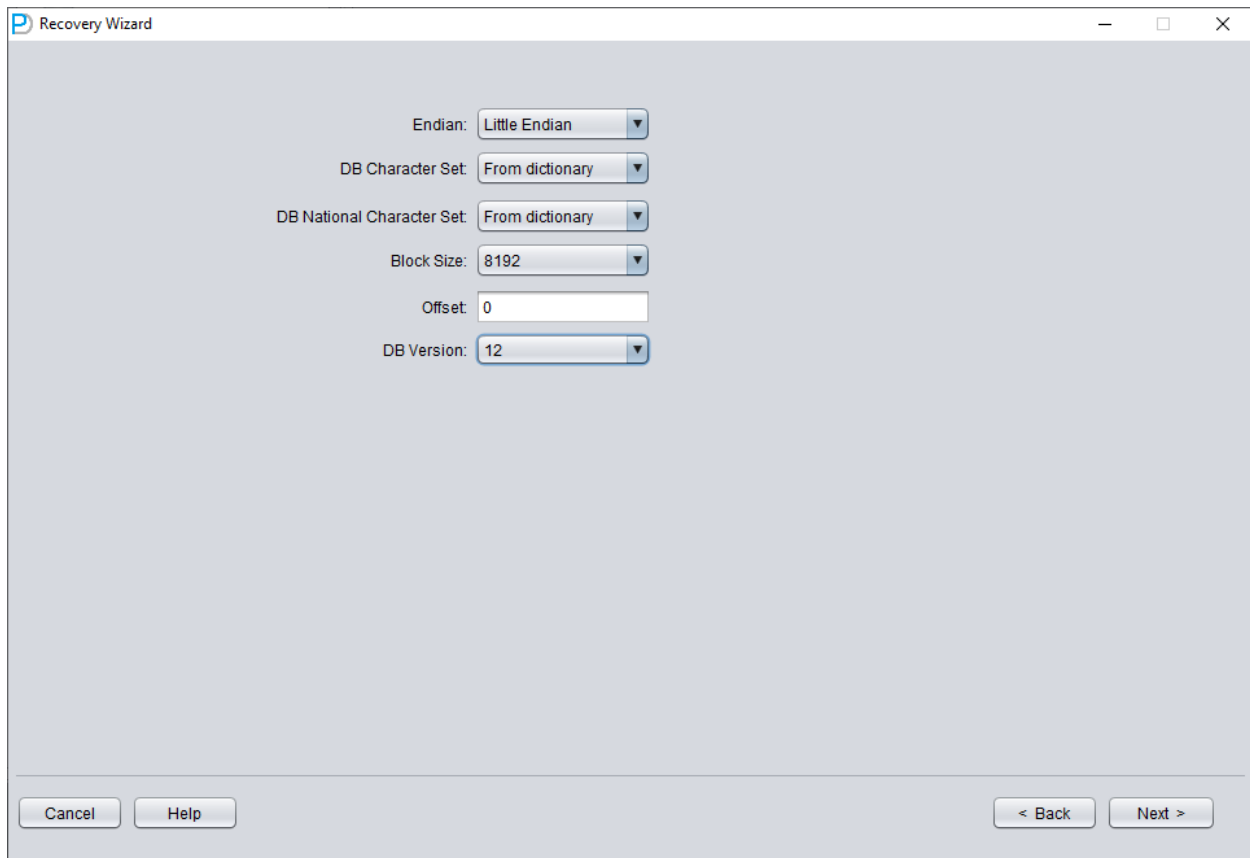
Notez que TEMP, UNDOTBS1, SYSAUX ne sont pas liés à notre travail de récupération, nous pouvons ignorer ces fichiers.

Nous commençons par lancer DBRECOVER, en utilisant le mode dictionnaire DICT-MODE :

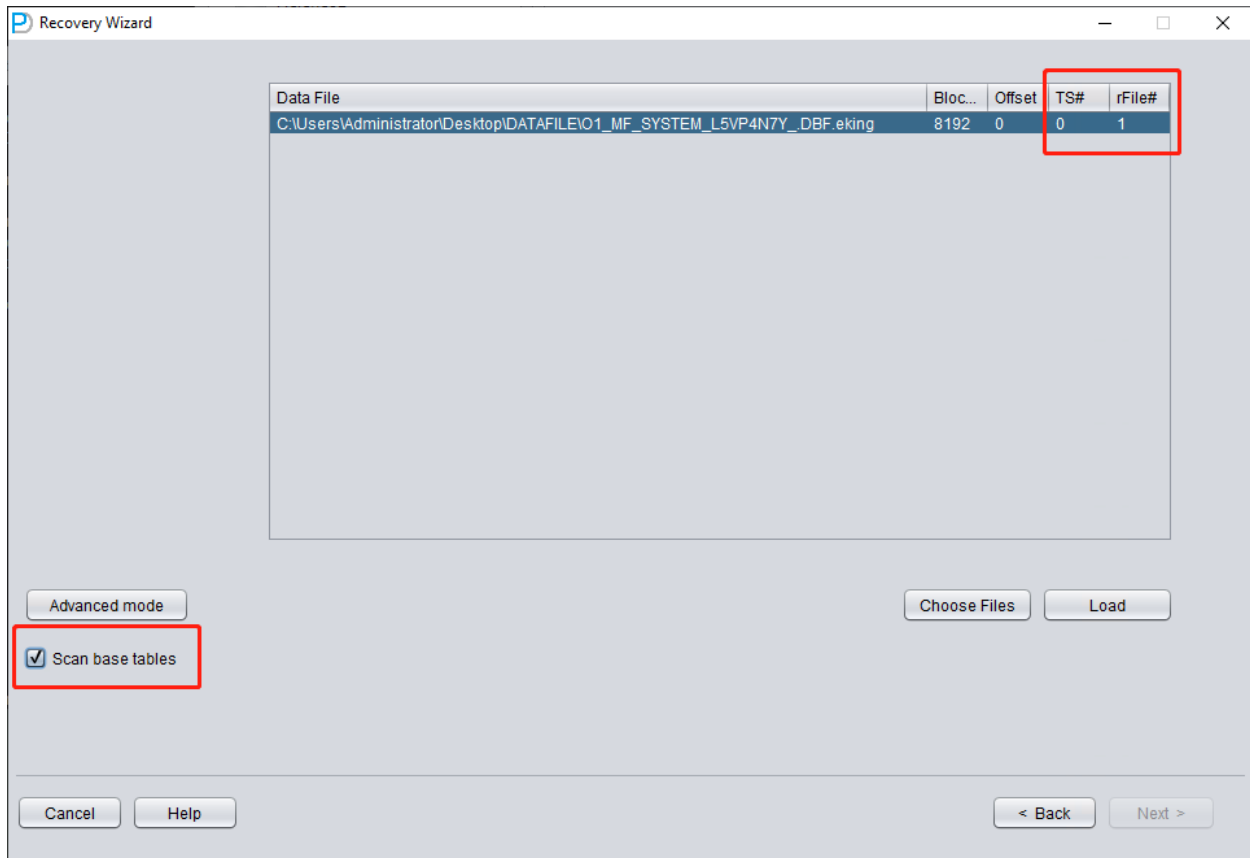




Choisissez DB VERSION en fonction de la situation réelle, pour les instances de version supérieure à 12c, comme 18c 19c, etc., choisissez 12.

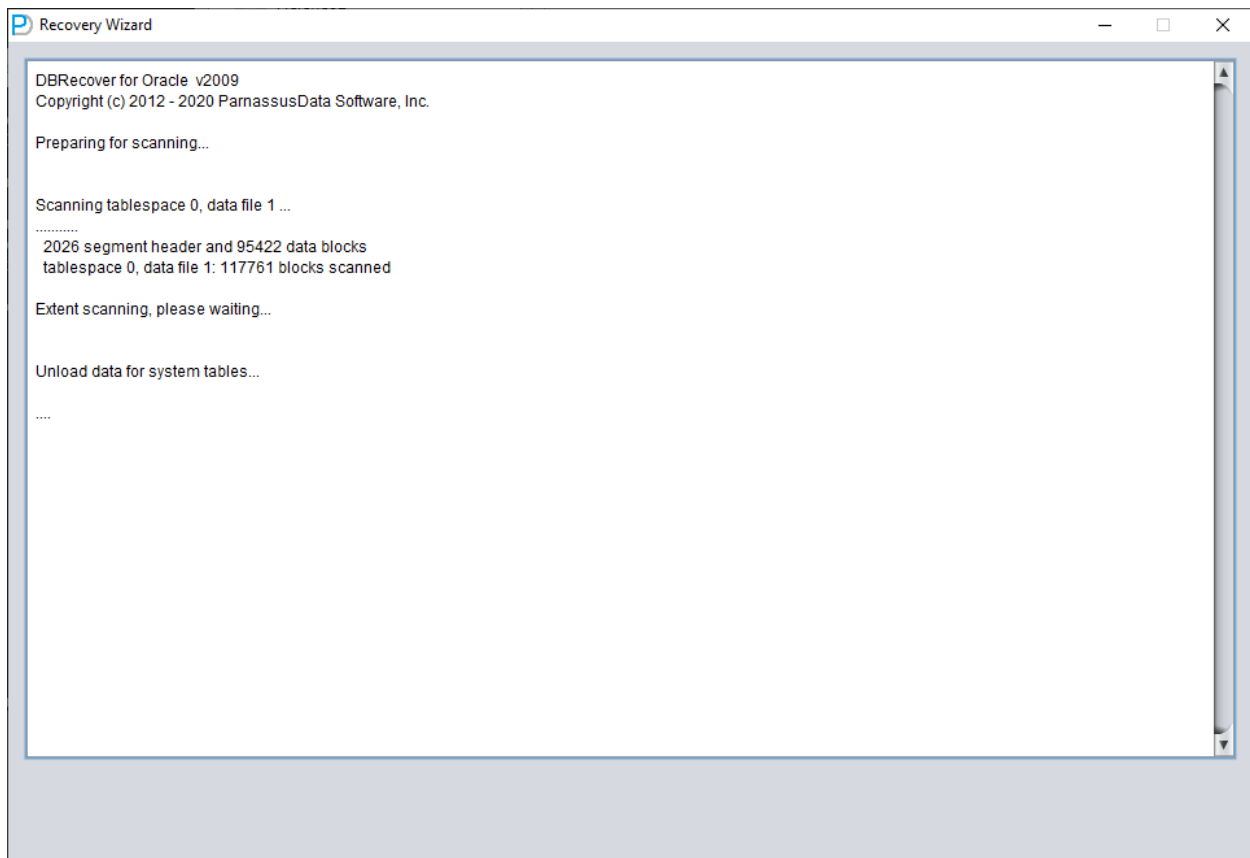


Incluez seulement SYSTEM01.DBF, et spécifiez son TS#=0 rFILE#=1 (notez que ceci est fixe).



La case "SCAN BASE TABLES" ci-dessus peut aider à faire face plus efficacement aux situations de dommages.

Après avoir cliqué sur le bouton LOAD, DBRECOVER va scanner l'ensemble de SYSTEM01.DBF et trouver les données de la table de base du dictionnaire de données.



Nous ouvrons le nœud de l'utilisateur SYS, et recherchons les deux tables de base TS\$ et FILE\$:

DBRecover for Oracle email: service@pamassusdata.com www.dbrecover.com Professional Oracle Database Disaster Recovery Version 2009

Start Options

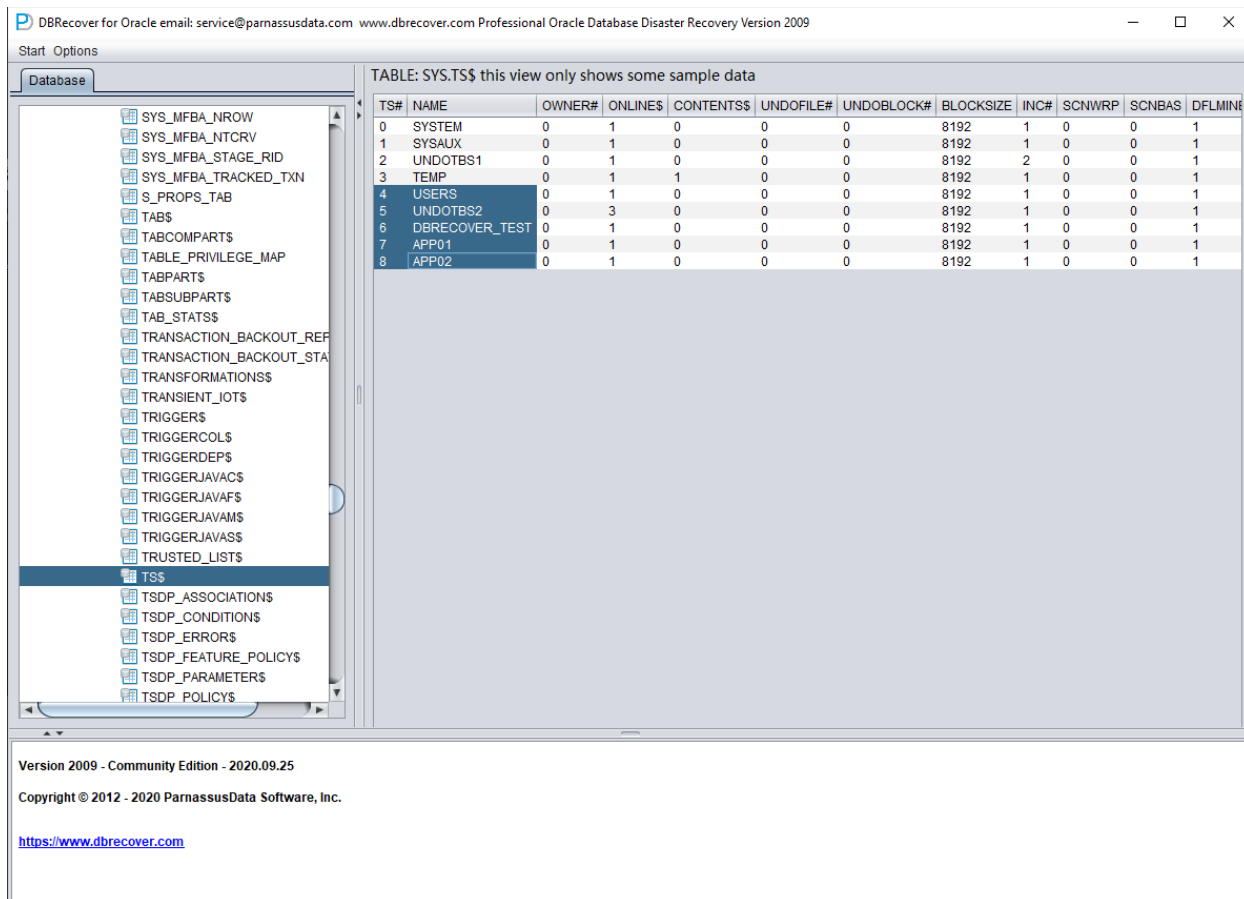
Database

- Database
 - DB_20230529123253
 - Users
 - GSMROOTUSER
 - PD
 - SCOTT
 - SYS
 - Tables
 - ACCESS\$
 - ACLMV\$
 - ACLMV\$REFLOG
 - ACLMVREFSTAT\$
 - ACLMVSUBTBL\$
 - ADMINAUTH\$
 - ADO_IMP_PARAMS\$
 - ADO_IMSEGSTAT\$
 - ADO_IMSEGTASKDETAILS\$
 - ADO_IMSTAT\$
 - ADO_IMTASKS\$
 - ALERT_QT
 - ALL_UNIFIED_AUDIT_ACTIONS
 - APPLY\$_AUTO_CDR_COLUMN
 - APPLY\$_BATCH_SQL_STATS
 - APPLY\$_CDR_INFO
 - APPLY\$CHANGE_HANDLERS
 - APPLY\$CONF_HDLR_COLUMN
 - APPLY\$CONSTRAINT_COLUMN

TABLE: SYS.TS\$ this view only shows some sample data

TS#	NAME	OWNER#	ONLINES	CONTENTSS	UNDOFILE#	UNDOBLOCK#	BLOCKSIZE	INC#	SCNWRP	SCNBAS	DFLMIN
0	SYSTEM	0	1	0	0	0	8192	1	0	0	1
1	SYS_AUX	0	1	0	0	0	8192	1	0	0	1
2	UNDOTBS1	0	1	0	0	0	8192	2	0	0	1
3	TEMP	0	1	1	0	0	8192	1	0	0	1
4	USERS	0	1	0	0	0	8192	1	0	0	1
5	UNDOTBS2	0	3	0	0	0	8192	1	0	0	1
6	DBRECOVER_TEST	0	1	0	0	0	8192	1	0	0	1
7	APP01	0	1	0	0	0	8192	1	0	0	1
8	APP02	0	1	0	0	0	8192	1	0	0	1

Version 2009 - Community Edition - 2020.09.25
 Copyright © 2012 - 2020 PamassusData Software, Inc.
<https://www.dbrecover.com>



La table TS\$ stocke les informations de l'espace de table, la colonne TS# est le numéro de l'espace de table, et nous pouvons obtenir les informations suivantes :

TS#	NAME
0	SYSTEM
1	SYSAUX
2	UNDOTBS1
3	TEMP
4	USERS
5	UNDOTBS2
6	DBRECOVER_TEST
7	APP01
8	APP02

C'est-à-dire que le TS# de l'espace de table APP01 est 7, et que le TS# de l'espace de table APP02 est 8.

La table FILE\$ stocke les informations sur les fichiers de données :

TABLE: SYS.FILE\$ this view only shows some sample data

FILE#	STATUS\$	BLOCKS	TS#	RELFIL#	MAXEXTEND	INC	CRSCNWRP	CRSCNBAS	OWNERINSTANCE	SPARE1	SPARE2	SPARE
1	2	64000	0	1	4194302	1280	0	8		4194306	0	
3	2	51200	1	3	4194302	1280	0	6336		12582914	0	
5	2	64000	6	5	0	0	0	2447818		20971522	0	
7	2	640	4	7	4194302	160	0	37099		29360130	0	
2	2	64000	7	2	0	0	0	3207824		8388610	0	
4	2	3200	2	4	4194302	640	0	1928280		16777218	0	
8	2	12800	7	8	0	0	0	3207853		33554434	0	
9	2	12800	7	9	0	0	0	3207872		37748738	0	
10	2	64000	8	10	0	0	0	3207900		41943042	0	
11	2	12800	8	11	0	0	0	3207935		46137346	0	
12	2	12800	8	12	0	0	0	3207955		50331650	0	

Version 2009 - Community Edition - 2020.09.25
 Copyright © 2012 - 2020 ParnassusData Software, Inc.
<https://www.dbrecover.com>

Ce dont nous avons besoin sont les deux colonnes TS# et RELFILE#

TS#	RELFIL#
0	1
1	3
6	5
4	7

7	2
2	4
7	8
7	9
8	10
8	11
8	12

En fusionnant les données des deux tableaux, nous obtenons :

TS#	RELFIL#	Nom de l'espace de table
0	1	SYSTEM
1	3	SYSAUX
6	5	DBRECOVER_TEST
4	7	USERS
7	2	APP01
2	4	UNDOTBS1
7	8	APP01
7	9	APP01
8	10	APP02
8	11	APP02
8	12	APP02

En supprimant les espaces de table inutiles SYSAUX, UNDOTBS1 et le SYSTEM déjà connu, il ne reste que :

TS#	RELFIL#	Nom de l'espace de table
6	5	DBRECOVER_TEST
4	7	USERS
7	2	APP01
7	8	APP01
7	9	APP01
8	10	APP02
8	11	APP02
8	12	APP02

Liste des noms de fichiers de données correspondants :

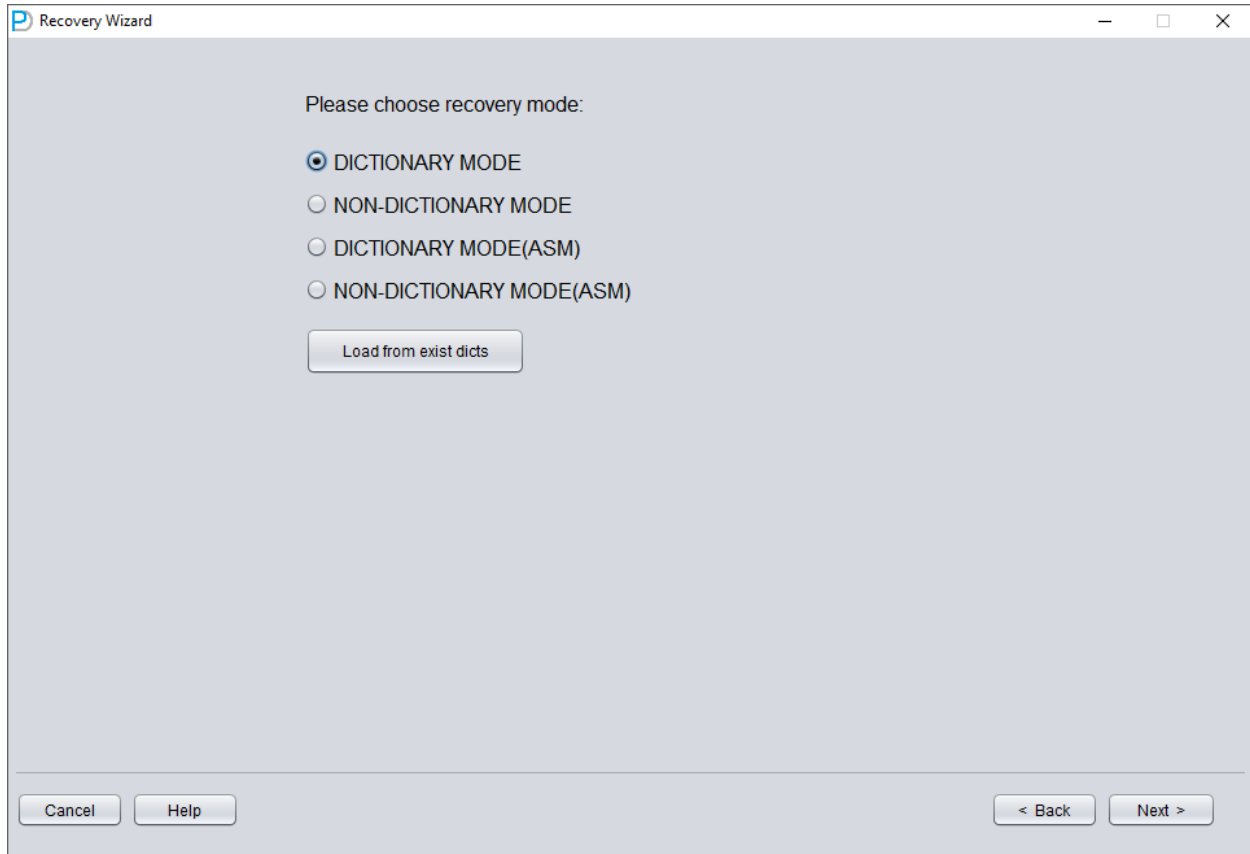
O1_MF_APP01_L782YY4Y_.DBF.eking
O1_MF_APP01_L782ZBM3_.DBF.eking
O1_MF_APP01_L782ZCP1_.DBF.eking
O1_MF_APP02_L782ZO7W_.DBF.eking
O1_MF_APP02_L7830DTG_.DBF.eking
O1_MF_APP02_L7830FJ6_.DBF.eking
O1_MF_DBRECOVE_L6G7B1Q3_.DBF.eking
O1_MF_USERS_L5VP67TJ_.DBF.eking

En comparant les deux tableaux ci-dessus, il n'est pas difficile de voir la correspondance. Pour les fichiers de données gérés par db_create_file_dest OMF, les multiples fichiers de données sous un espace de table peuvent être triés par leur nom de fichier, qui est dans l'ordre avec le tri RELFILE#. Pour les noms de fichiers gérés par l'utilisateur (c'est-à-dire sans utiliser OMF), ils seront généralement nommés APP01{XX} (comme APP0101, APP0102) pour faciliter la gestion, et leur correspondance peut également être obtenue.

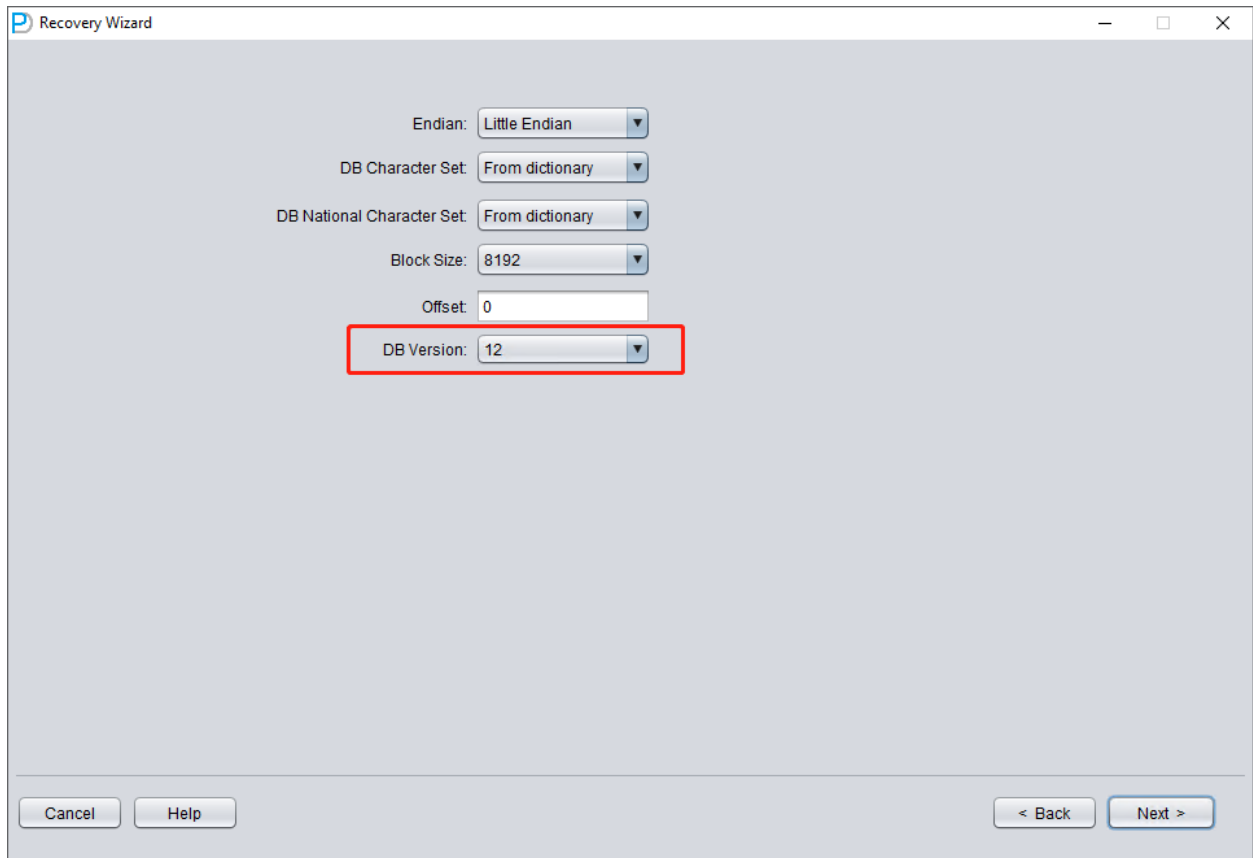
Par conjecture, nous avons obtenu le tableau d'informations complet :

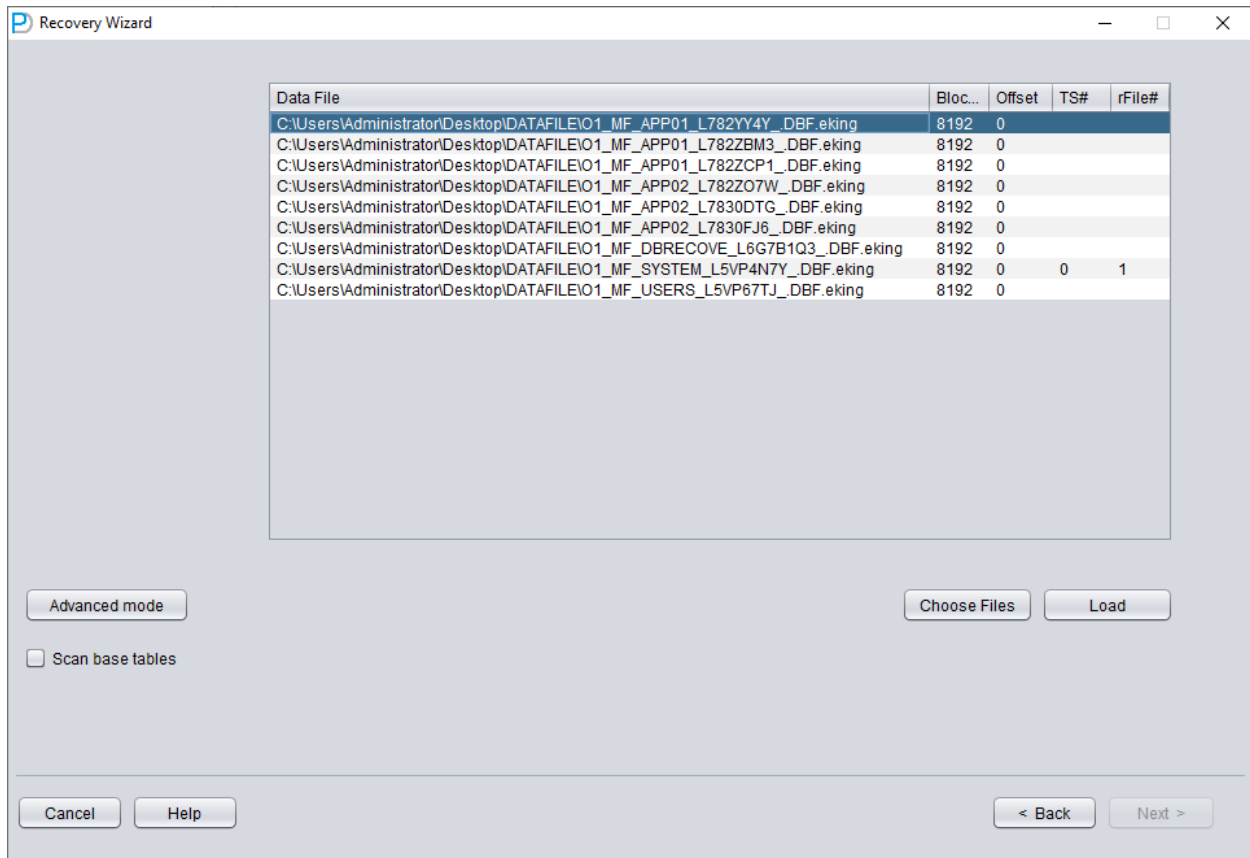
TS#	RFILE#	Nom de l'espace de table	NOM DE FICHIER
6	5	DBRECOVER_TEST	O1_MF_DBRECOVE_L6G7B1Q3_.DBF.eking
4	7	USERS	O1_MF_USERS_L5VP67TJ_.DBF.eking
7	2	APP01	O1_MF_APP01_L782YY4Y_.DBF.eking
7	8	APP01	O1_MF_APP01_L782ZBM3_.DBF.eking
7	9	APP01	O1_MF_APP01_L782ZCP1_.DBF.eking
8	10	APP02	O1_MF_APP02_L782ZO7W_.DBF.eking
8	11	APP02	O1_MF_APP02_L7830DTG_.DBF.eking
8	12	APP02	O1_MF_APP02_L7830FJ6_.DBF.eking

Réouvrez DBRECOVER et entrez en mode dictionnaire :



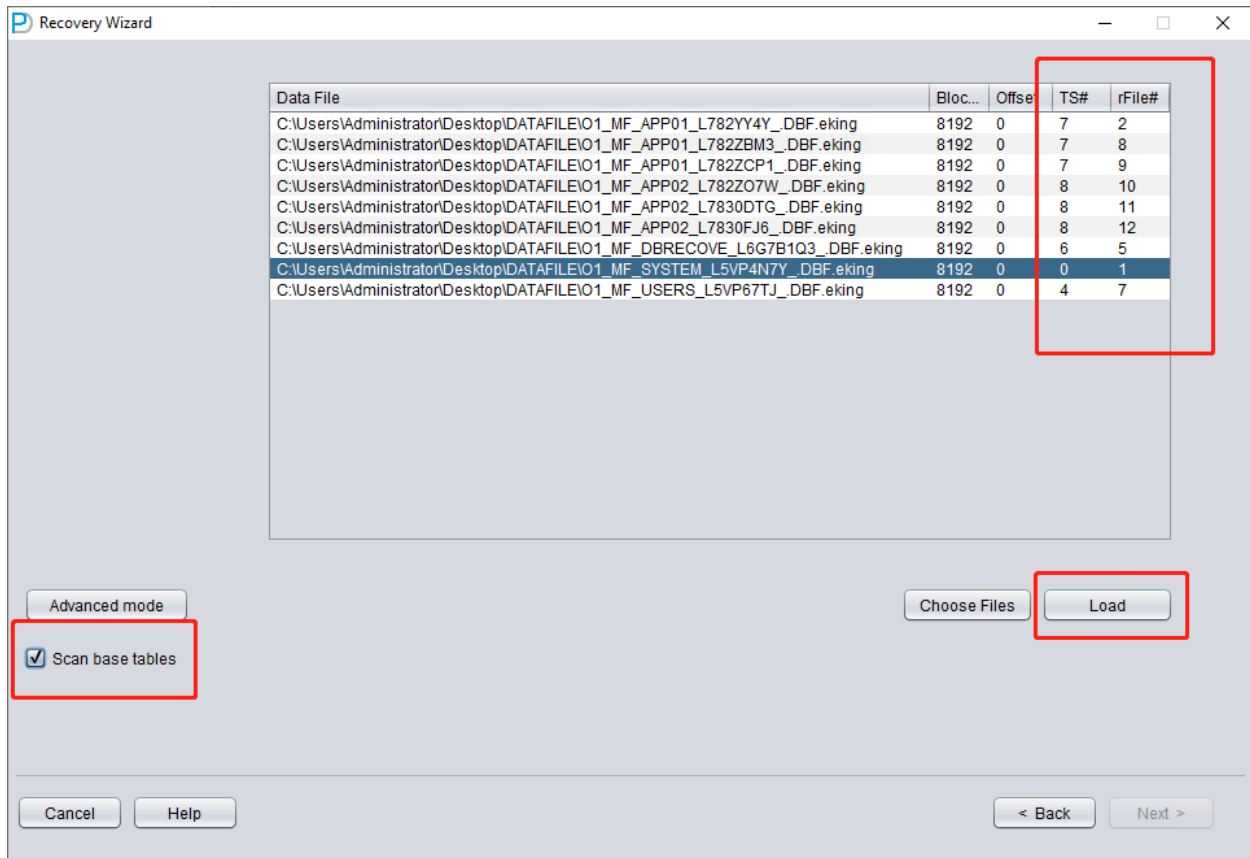
Vous devez toujours choisir la version de la base de données DB VERSION.



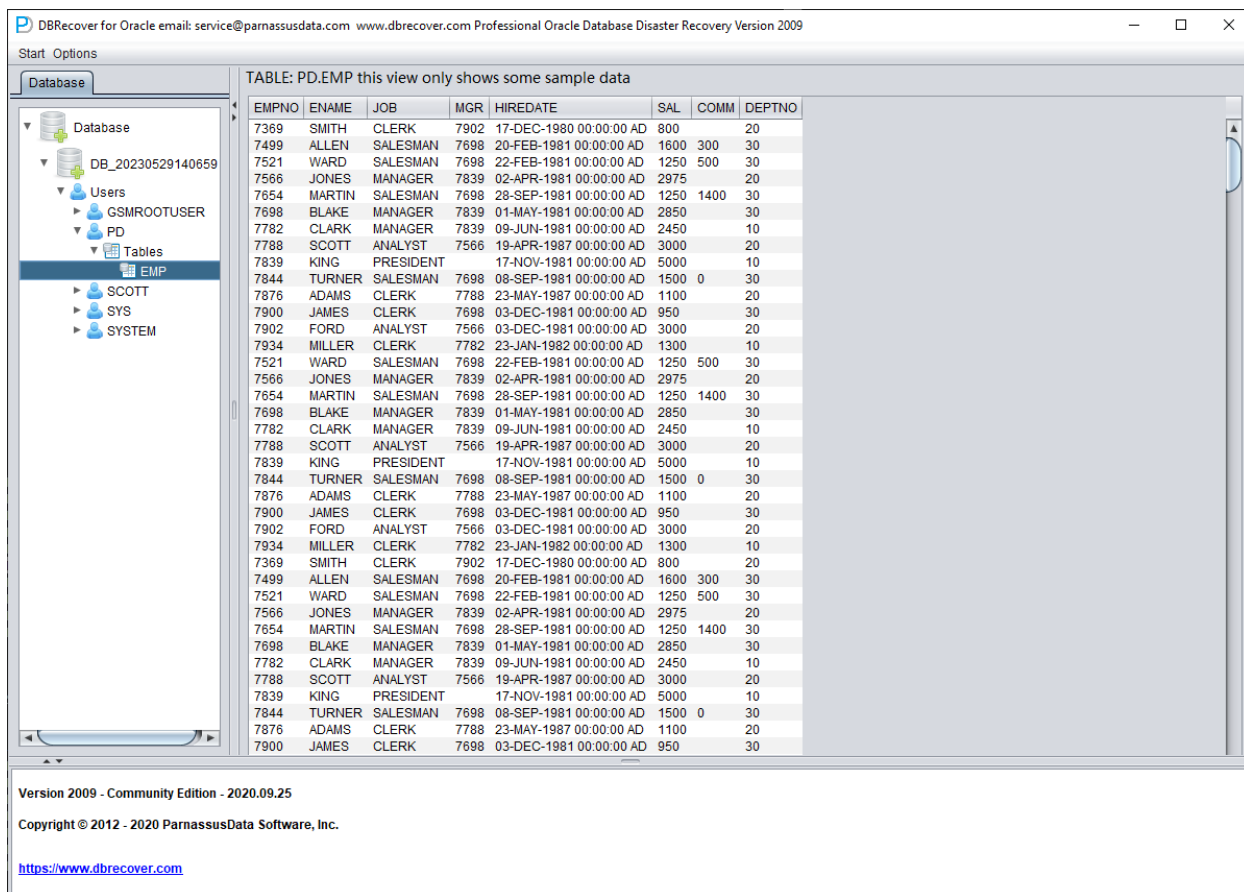


Ajoutez tous les fichiers de données nécessaires (tous les fichiers qui peuvent contenir des données utilisateur, UNDOTBS1, TEMP, SYSAUX ne doivent pas être ajoutés), veillez à ne pas oublier SYSTEM01.DBF (doit être ajouté).

Selon le tableau que nous avons préparé précédemment, remplissez les informations TS# et RFILE# ici :



Si vous avez correctement rempli les informations et que le niveau de dommage du cryptage n'est pas élevé, vous devriez être en mesure de lire directement les données :



Comme les caractéristiques des virus de cryptage varient, vous pouvez rencontrer plus de problèmes lors de la mise en œuvre réelle. N'hésitez pas à nous contacter par e-mail pour discuter de vos problèmes : service@pamassusdata.com

Scénario de récupération 4 : Récupération des lignes de données supprimées par erreur avec DELETE FROM TABLE

Un développeur de la société D a exécuté par erreur un script de suppression de données de l'environnement de test sur l'environnement de production (PROD DATABASE) ; toutes les données d'une table ont été supprimées avec DELETE.

Dans ce scénario, nous pouvons utiliser DBRECOVER pour récupérer les lignes de données supprimées.

Cependant, l'utilisateur doit d'abord effectuer les opérations suivantes pour protéger au maximum les données contre l'écrasement :

1. Mettre l'espace de table où se trouve la table en mode lecture seule READ ONLY, avec la commande : ALTER TABLESPACE {TABLESPACE_NAME} READ ONLY
2. Arrêter l'instance de la base de données : SHUTDOWN IMMEDIATE

L'utilisateur peut choisir l'une ou l'autre de ces deux solutions.

Reproduire le scénario :

```
SQL> select count(*) from pd.emp;
COUNT(*)
-----
114688
SQL>
```

```
SQL> delete from pd.emp;
114688 rows deleted.
SQL> commit;
Commit complete.
SQL> alter system checkpoint;
System altered.
SQL> select count(*) from pd.emp;
COUNT(*)
-----
0
```

Avant de commencer la récupération, nous mettons d'abord l'espace de table en lecture seule pour protéger l'environnement de récupération :

```
SQL> select tablespace_name from dba_segments where owner='PD' and segment_name='EMP';
```

```
TABLESPACE_NAME
```

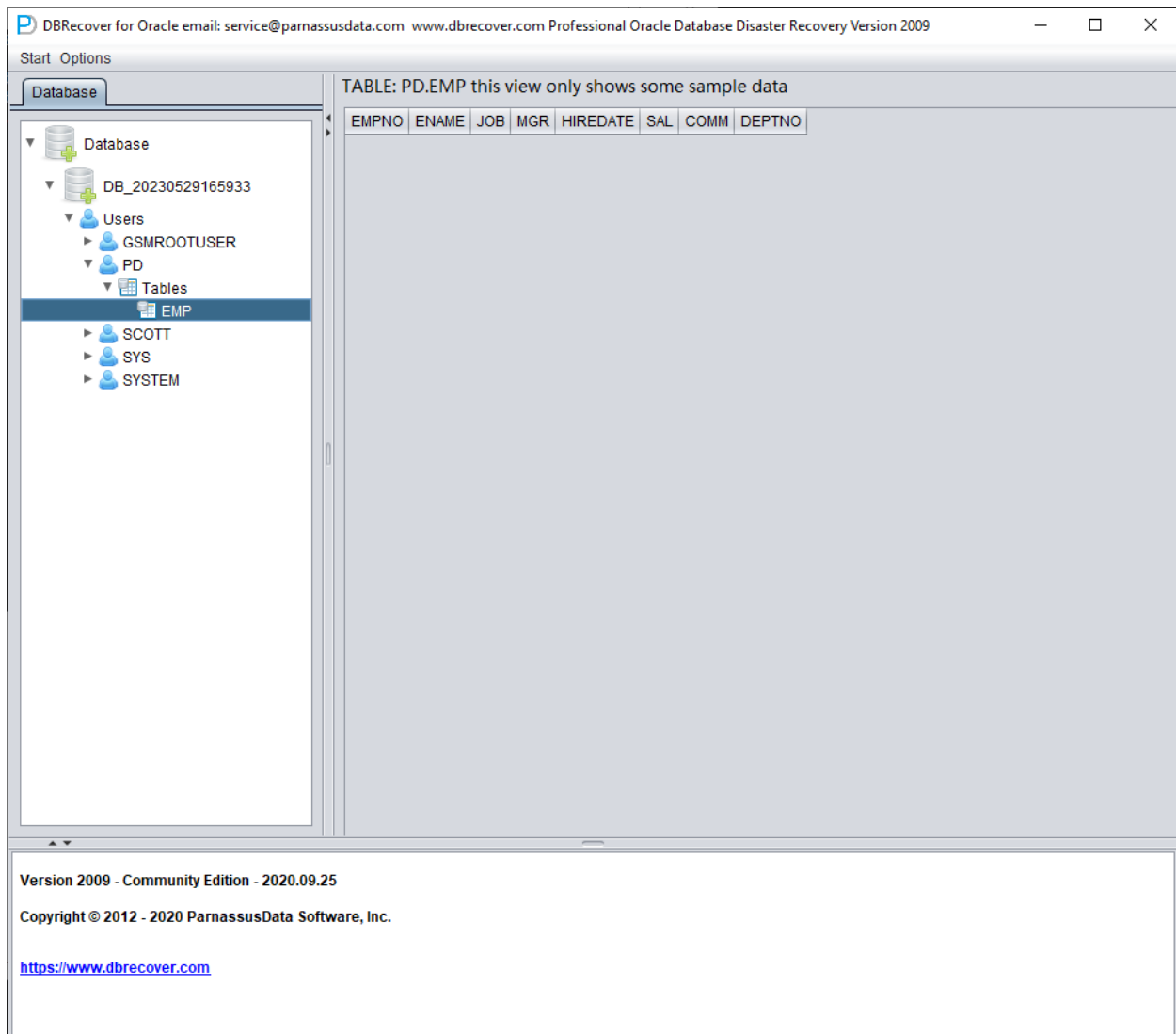
```
-----
```

```
DBRECOVER_TEST
```

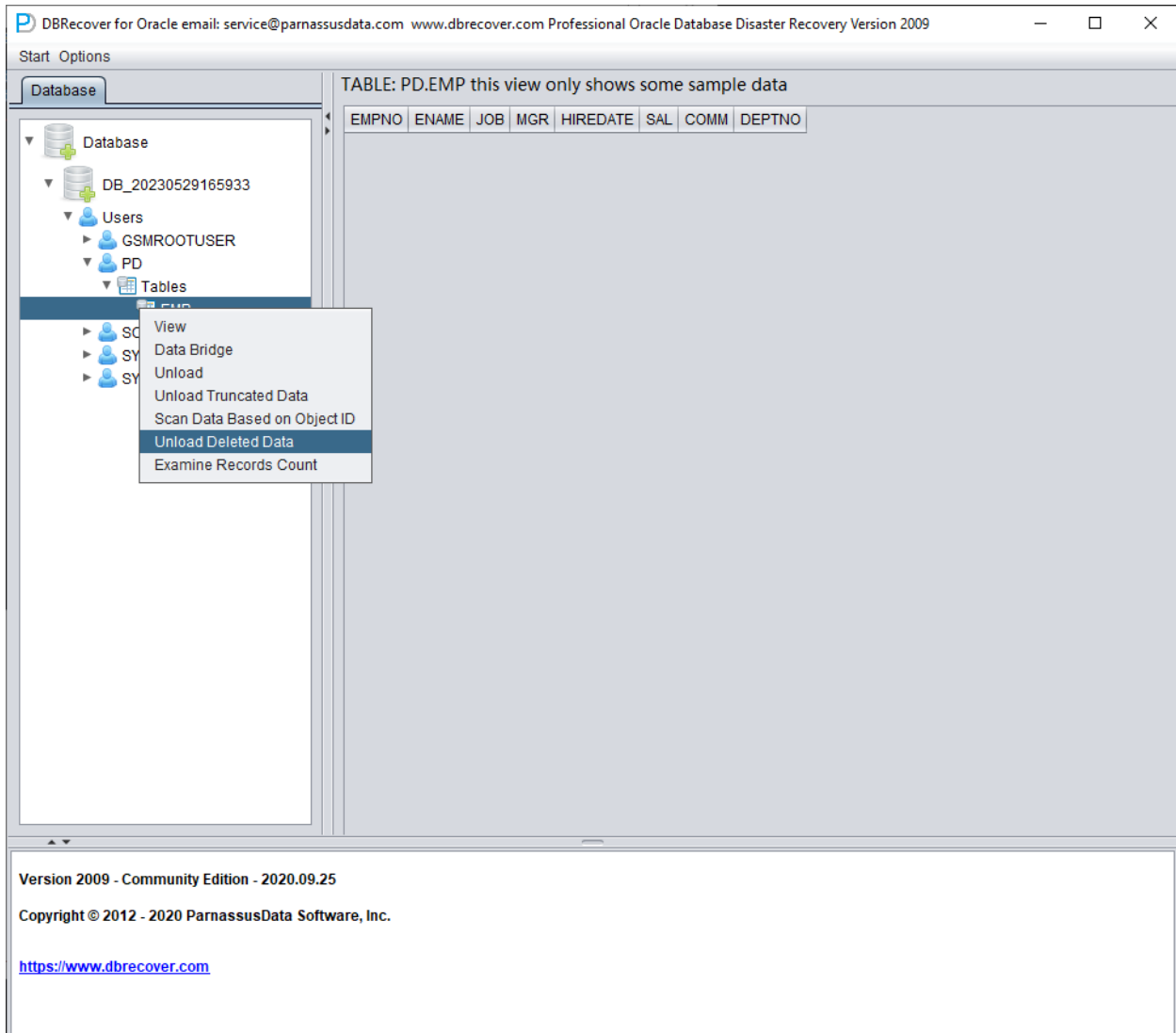
```
SQL> alter tablespace DBRECOVER_TEST read only;
```

```
Tablespace altered.
```

Démarrez DBRECOVER, choisissez le mode dictionnaire et ajoutez tous les fichiers de données disponibles :

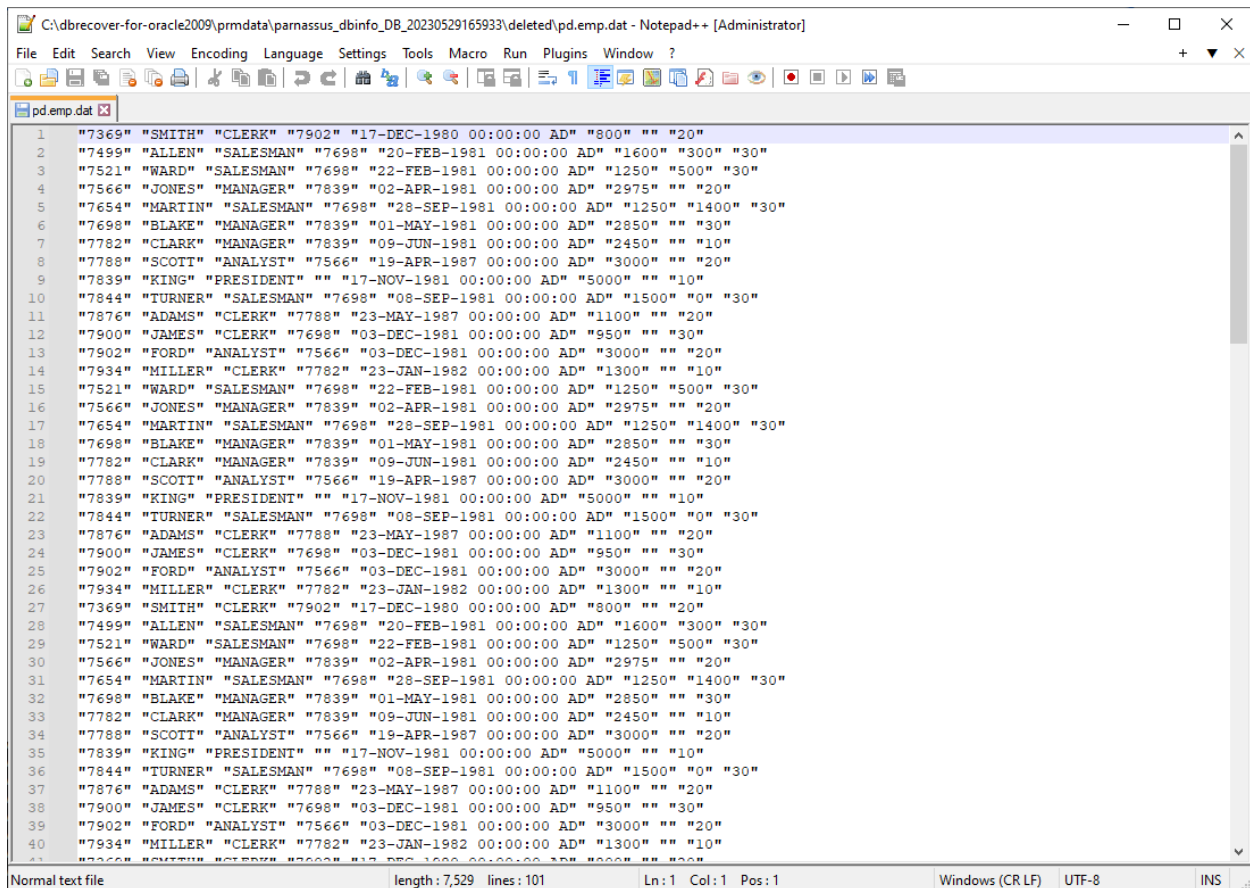


En regardant les données de la table d'exemple, on peut voir qu'elle est vide. Sélectionnez la table, faites un clic droit et choisissez Unload Deleted Data.



Sans une licence d'entreprise valide, la fonction UNLOAD DELETED DATA est limitée à 100 lignes de données par table.

Les données extraites sont stockées dans le chemin indiqué par la fenêtre contextuelle :



```
C:\dbrecover-for-oracle2009\prmdata\parnassus_dbinfo_DB_20230529165933\deleted\pd.emp.dat - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
pd.emp.dat
1 "7369" "SMITH" "CLERK" "7902" "17-DEC-1980 00:00:00 AD" "800" "" "20"
2 "7499" "ALLEN" "SALESMAN" "7698" "20-FEB-1981 00:00:00 AD" "1600" "300" "30"
3 "7521" "WARD" "SALESMAN" "7698" "22-FEB-1981 00:00:00 AD" "1250" "500" "30"
4 "7566" "JONES" "MANAGER" "7839" "02-APR-1981 00:00:00 AD" "2975" "" "20"
5 "7654" "MARTIN" "SALESMAN" "7698" "28-SEP-1981 00:00:00 AD" "1250" "1400" "30"
6 "7698" "BLAKE" "MANAGER" "7839" "01-MAY-1981 00:00:00 AD" "2850" "" "30"
7 "7782" "CLARK" "MANAGER" "7839" "09-JUN-1981 00:00:00 AD" "2450" "" "10"
8 "7788" "SCOTT" "ANALYST" "7566" "19-APR-1987 00:00:00 AD" "3000" "" "20"
9 "7839" "KING" "PRESIDENT" "" "17-NOV-1981 00:00:00 AD" "5000" "" "10"
10 "7844" "TURNER" "SALESMAN" "7698" "08-SEP-1981 00:00:00 AD" "1500" "0" "30"
11 "7876" "ADAMS" "CLERK" "7788" "23-MAY-1987 00:00:00 AD" "1100" "" "20"
12 "7900" "JAMES" "CLERK" "7698" "03-DEC-1981 00:00:00 AD" "950" "" "30"
13 "7902" "FORD" "ANALYST" "7566" "03-DEC-1981 00:00:00 AD" "3000" "" "20"
14 "7934" "MILLER" "CLERK" "7782" "23-JAN-1982 00:00:00 AD" "1300" "" "10"
15 "7521" "WARD" "SALESMAN" "7698" "22-FEB-1981 00:00:00 AD" "1250" "500" "30"
16 "7566" "JONES" "MANAGER" "7839" "02-APR-1981 00:00:00 AD" "2975" "" "20"
17 "7654" "MARTIN" "SALESMAN" "7698" "28-SEP-1981 00:00:00 AD" "1250" "1400" "30"
18 "7698" "BLAKE" "MANAGER" "7839" "01-MAY-1981 00:00:00 AD" "2850" "" "30"
19 "7782" "CLARK" "MANAGER" "7839" "09-JUN-1981 00:00:00 AD" "2450" "" "10"
20 "7788" "SCOTT" "ANALYST" "7566" "19-APR-1987 00:00:00 AD" "3000" "" "20"
21 "7839" "KING" "PRESIDENT" "" "17-NOV-1981 00:00:00 AD" "5000" "" "10"
22 "7844" "TURNER" "SALESMAN" "7698" "08-SEP-1981 00:00:00 AD" "1500" "0" "30"
23 "7876" "ADAMS" "CLERK" "7788" "23-MAY-1987 00:00:00 AD" "1100" "" "20"
24 "7900" "JAMES" "CLERK" "7698" "03-DEC-1981 00:00:00 AD" "950" "" "30"
25 "7902" "FORD" "ANALYST" "7566" "03-DEC-1981 00:00:00 AD" "3000" "" "20"
26 "7934" "MILLER" "CLERK" "7782" "23-JAN-1982 00:00:00 AD" "1300" "" "10"
27 "7369" "SMITH" "CLERK" "7902" "17-DEC-1980 00:00:00 AD" "800" "" "20"
28 "7499" "ALLEN" "SALESMAN" "7698" "20-FEB-1981 00:00:00 AD" "1600" "300" "30"
29 "7521" "WARD" "SALESMAN" "7698" "22-FEB-1981 00:00:00 AD" "1250" "500" "30"
30 "7566" "JONES" "MANAGER" "7839" "02-APR-1981 00:00:00 AD" "2975" "" "20"
31 "7654" "MARTIN" "SALESMAN" "7698" "28-SEP-1981 00:00:00 AD" "1250" "1400" "30"
32 "7698" "BLAKE" "MANAGER" "7839" "01-MAY-1981 00:00:00 AD" "2850" "" "30"
33 "7782" "CLARK" "MANAGER" "7839" "09-JUN-1981 00:00:00 AD" "2450" "" "10"
34 "7788" "SCOTT" "ANALYST" "7566" "19-APR-1987 00:00:00 AD" "3000" "" "20"
35 "7839" "KING" "PRESIDENT" "" "17-NOV-1981 00:00:00 AD" "5000" "" "10"
36 "7844" "TURNER" "SALESMAN" "7698" "08-SEP-1981 00:00:00 AD" "1500" "0" "30"
37 "7876" "ADAMS" "CLERK" "7788" "23-MAY-1987 00:00:00 AD" "1100" "" "20"
38 "7900" "JAMES" "CLERK" "7698" "03-DEC-1981 00:00:00 AD" "950" "" "30"
39 "7902" "FORD" "ANALYST" "7566" "03-DEC-1981 00:00:00 AD" "3000" "" "20"
40 "7934" "MILLER" "CLERK" "7782" "23-JAN-1982 00:00:00 AD" "1300" "" "10"
41 "7369" "SMITH" "CLERK" "7902" "17-DEC-1980 00:00:00 AD" "800" "" "20"
Normal text file length: 7,529 lines: 101 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS
```

L'utilisateur doit vérifier le résultat de la récupération lui-même et utiliser des outils tels que SQLLDR ou SQLDEVELOPER pour insérer les données textuelles dans la base de données.

Scénario de récupération 5 : récupération d'une table tronquée par erreur

Un membre du personnel de maintenance de l'entreprise D a tronqué par erreur toutes les données d'une table en confondant la base de données de production avec la base de données de test. Le DBA a tenté de récupérer mais a constaté que la dernière sauvegarde n'était pas utilisable, il ne pouvait donc pas récupérer les enregistrements de cette table à partir de la sauvegarde. Le DBA a alors décidé d'utiliser DBRECOVER pour récupérer les données qui avaient été tronquées.

Dans cet environnement, tous les fichiers de la base de données sont disponibles et sains, l'utilisateur a seulement besoin de charger les fichiers de données de l'espace de table SYSTEM et les fichiers de données de la table tronquée dans le mode dictionnaire, par exemple :

```
SQL> select count(*) From pd.salgrade;
```

```
COUNT(*)
```

```
-----
```

```
655360
```

```
SQL> select tablespace_name from dba_segments where owner='PD' and segment_name='SALGRADE';
```

```
TABLESPACE_NAME
```

```
-----
```

```
APP01
```

```
SQL> truncate table pd.salgrade;
```

```
Table truncated.
```

```
SQL>
```

```
SQL> alter system checkpoint;
```

```
System altered.
```

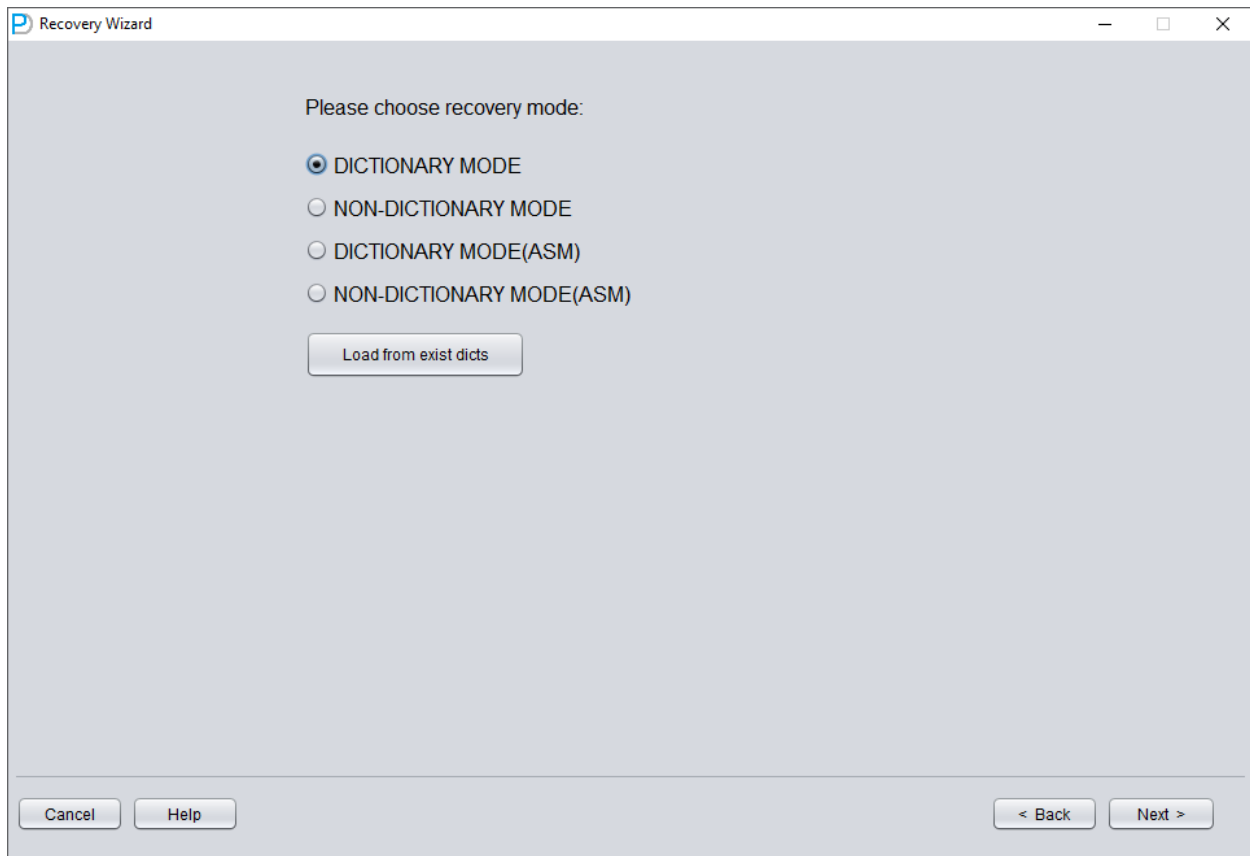
```
SQL> select count(*) from pd.salgrade;
```

```
COUNT(*)
```

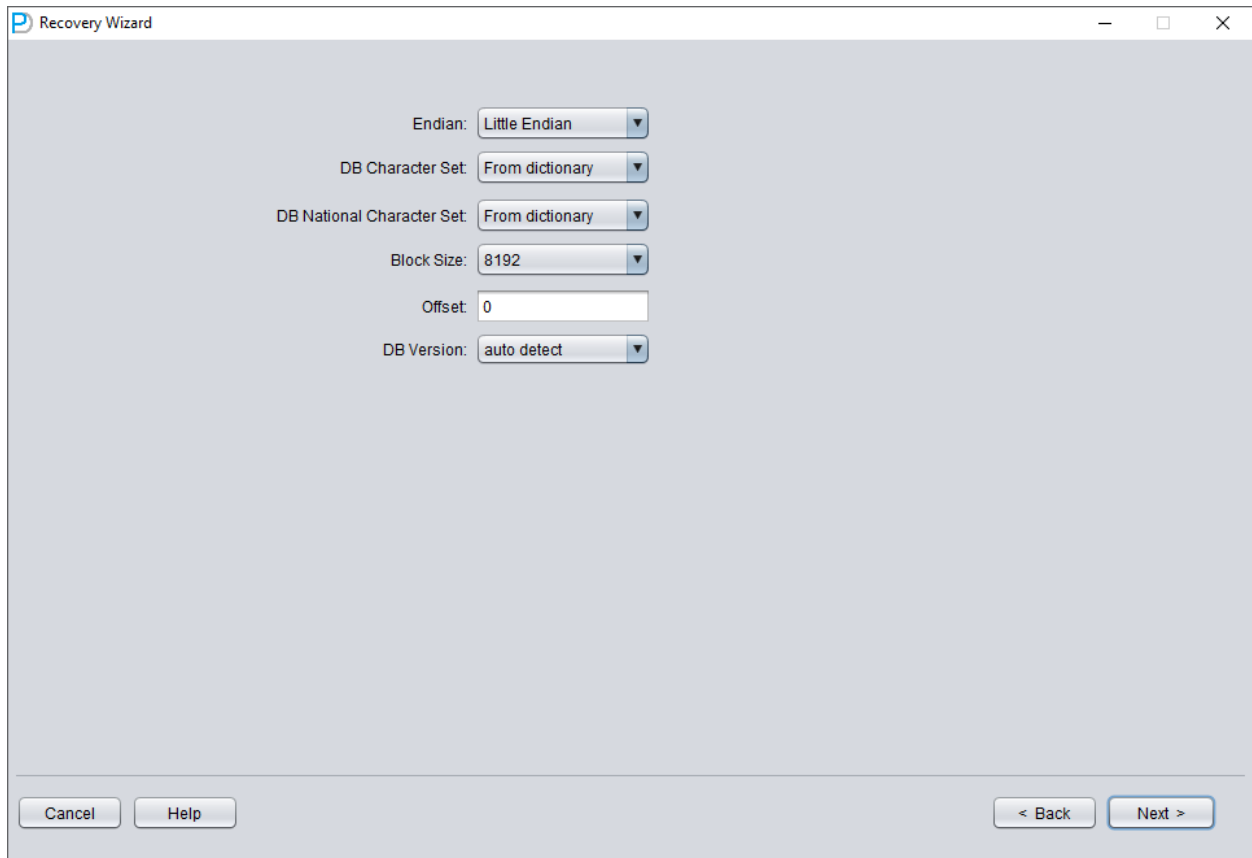
```
-----
```

```
0
```

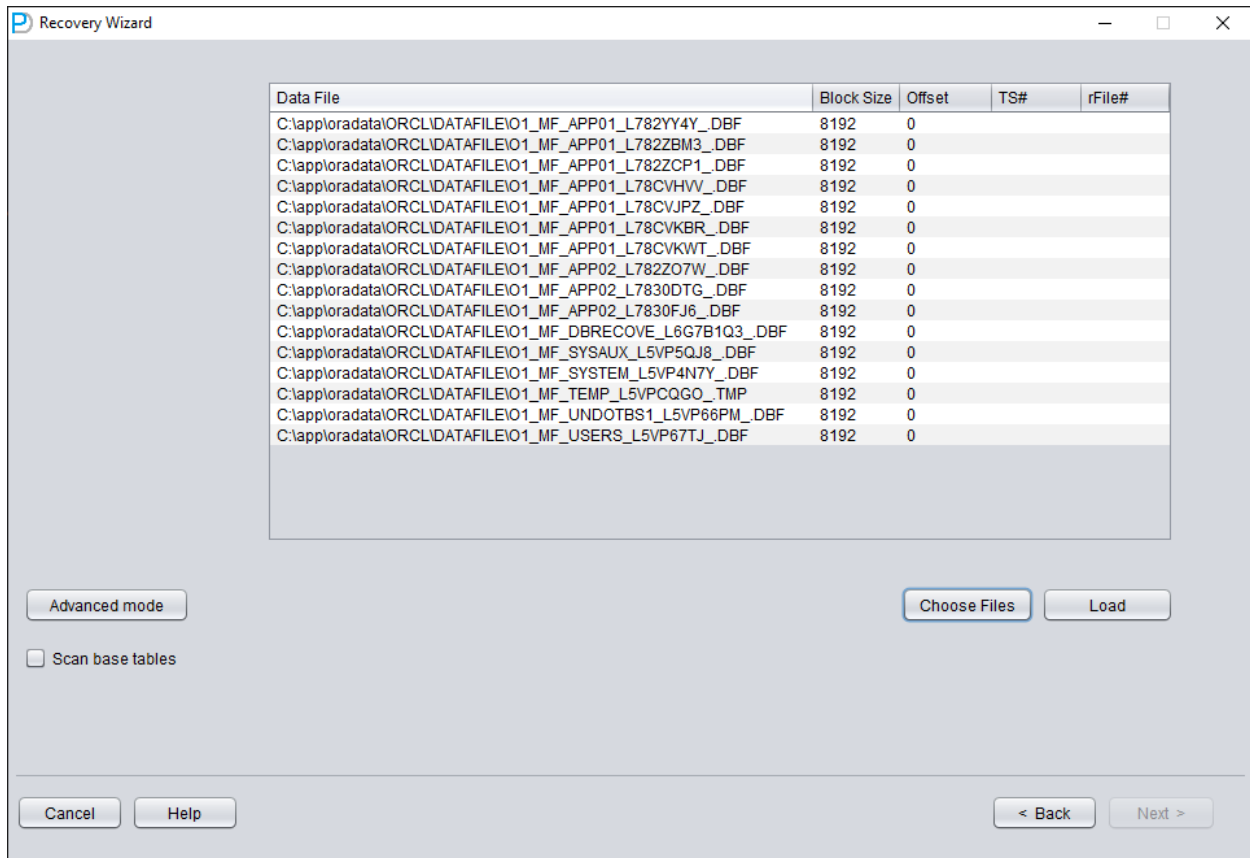
Dans ce scénario de TRUNCATE, le stockage ASM n'a pas été utilisé, donc il suffit de choisir le mode dictionnaire "Dictionary Mode" :



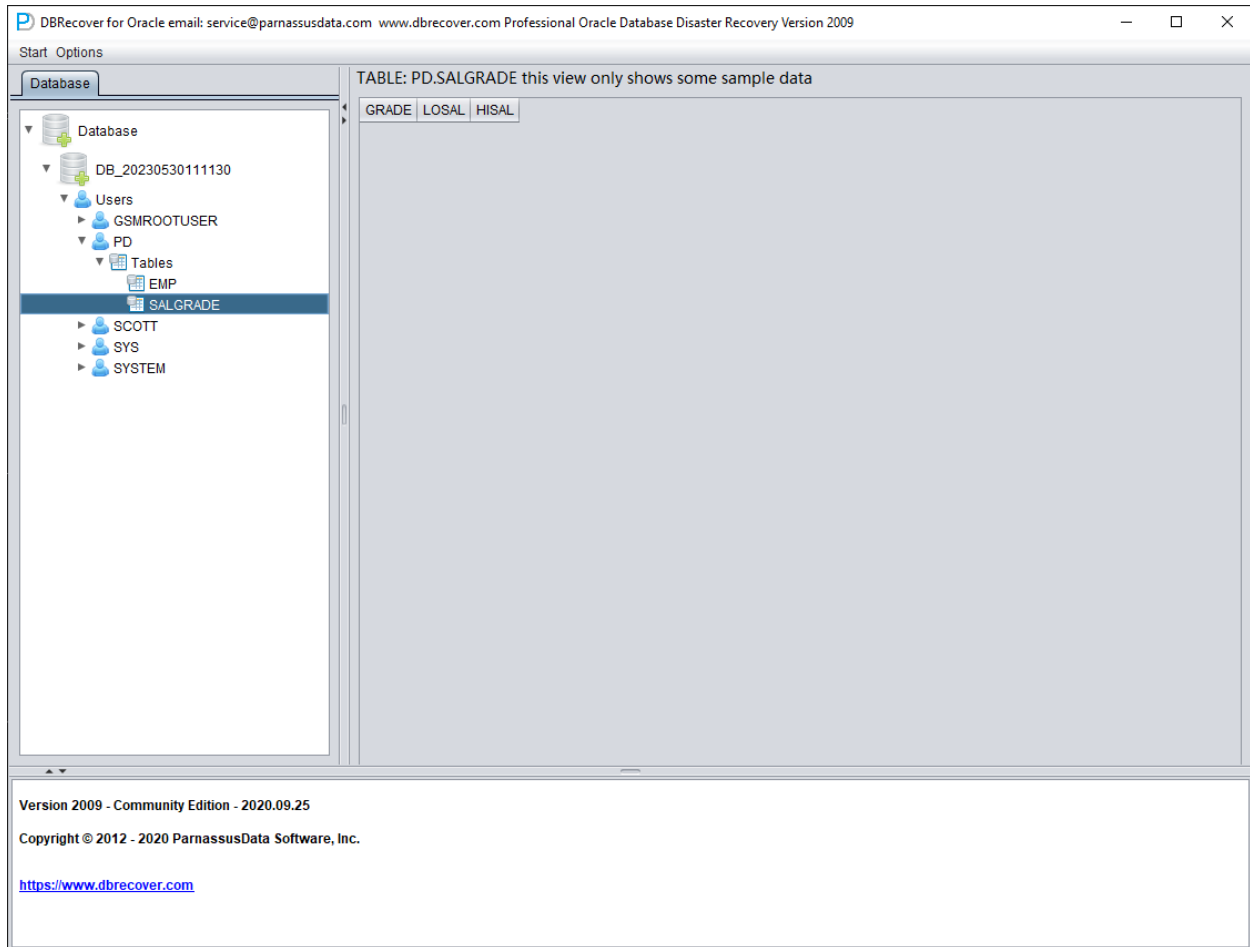
Dans la plupart des cas, il n'est pas nécessaire de modifier aucun paramètre :



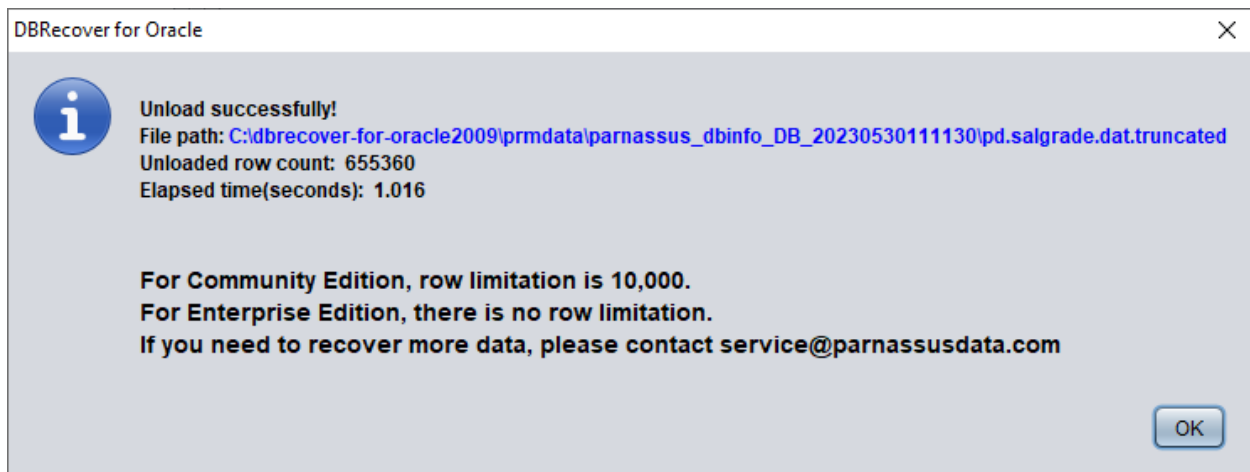
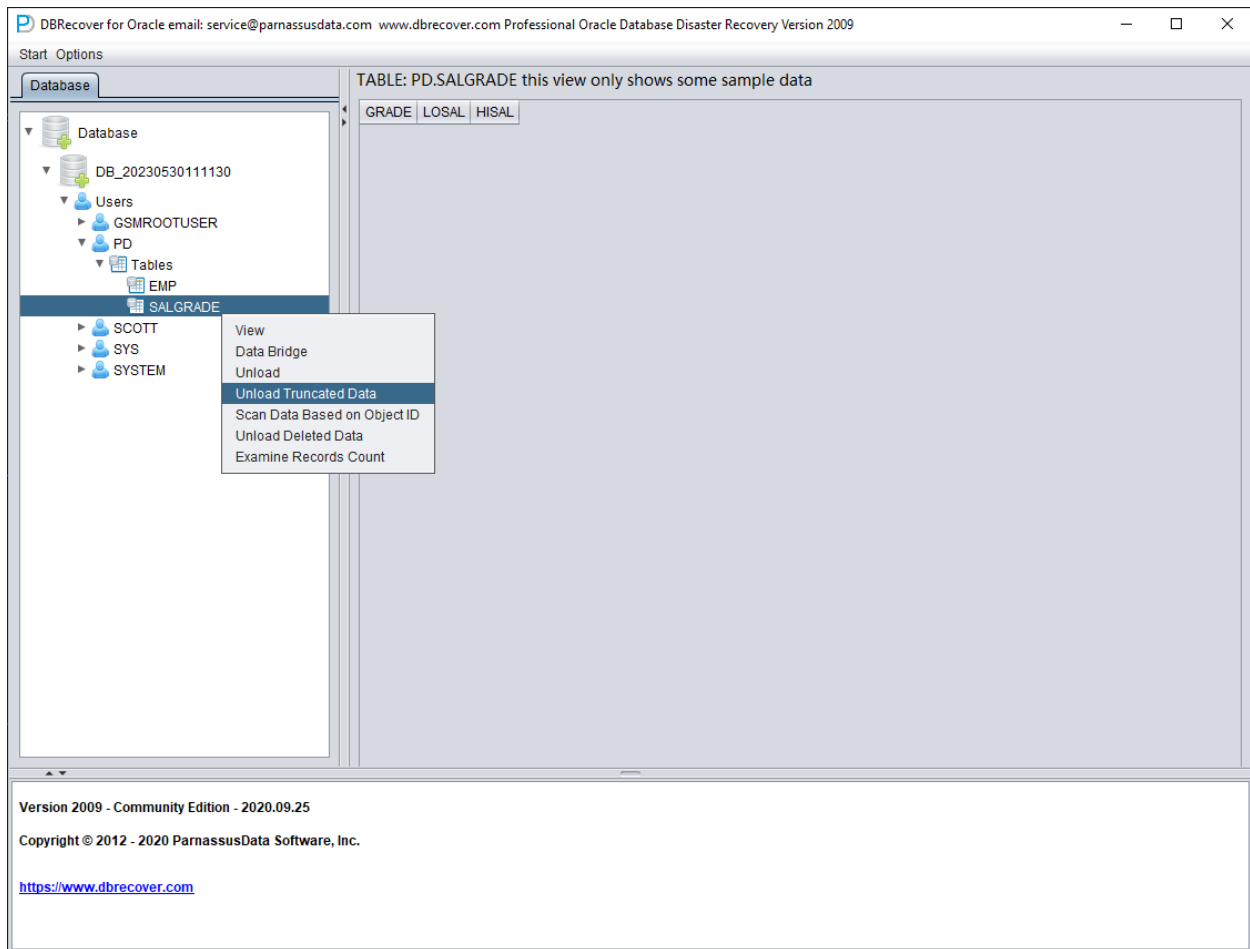
Ajoutez tous les fichiers de données disponibles :



Ouvrez USERS, vous pouvez voir plusieurs noms d'utilisateur, par exemple, si l'utilisateur a besoin de récupérer une table sous le schéma PD, ouvrez PD et double-cliquez sur le nom de la table :



Comme cette table a déjà été tronquée, double-cliquer ne montre pas de données, alors faites un clic droit sur la table et choisissez Unload truncated data :



DBRECOVER essaiera de balayer l'espace de la table et d'extraire les données qui ont été tronquées : comme le montre l'image ci-dessus, 655360 enregistrements complets ont été

extraits de la table qui avait été TRUNCATE et sont stockés dans le chemin indiqué par l'invite.

L'utilisateur peut vérifier le fichier DAT pour confirmer le résultat de la récupération.

La clé pour récupérer les données tronquées est de confirmer le DATA_OBJECT_ID de la table avant qu'elle ne soit tronquée, comme dans cet exemple :

```
SQL> select object_id ,data_object_id from dba_objects where owner='PD' and object_name='SALGRADE';  
OBJECT_ID DATA_OBJECT_ID  
-----  
76112 76113
```

Avant que TRUNCATE ne se produise, l'OBJECT_ID et le DATA_OBJECT_ID de cette table étaient tous deux 76612. Après TRUNCATE, le DATA_OBJECT_ID a changé.

Ainsi, l'original DATA_OBJECT_ID ici est 76612 ; mais si une table a été tronquée de nombreuses fois et que vous devez récupérer les données avant la dernière TRUNCATE, vous ne pouvez pas simplement utiliser l'OBJECT_ID pour deviner l'original DATA_OBJECT_ID.

Vous pouvez utiliser des techniques comme la requête flashback, la recherche de dictionnaire, l'exploration de logs, etc. pour déterminer le DATA_OBJECT_ID ; voici un exemple de requête flashback :

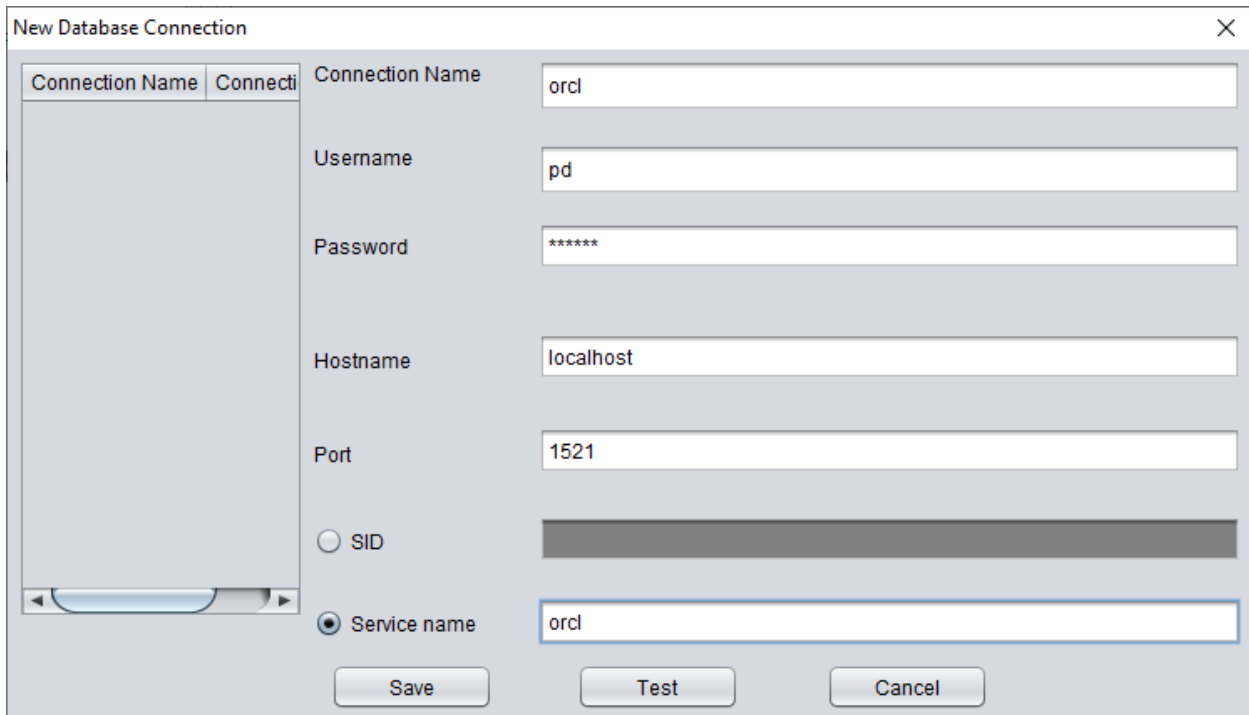
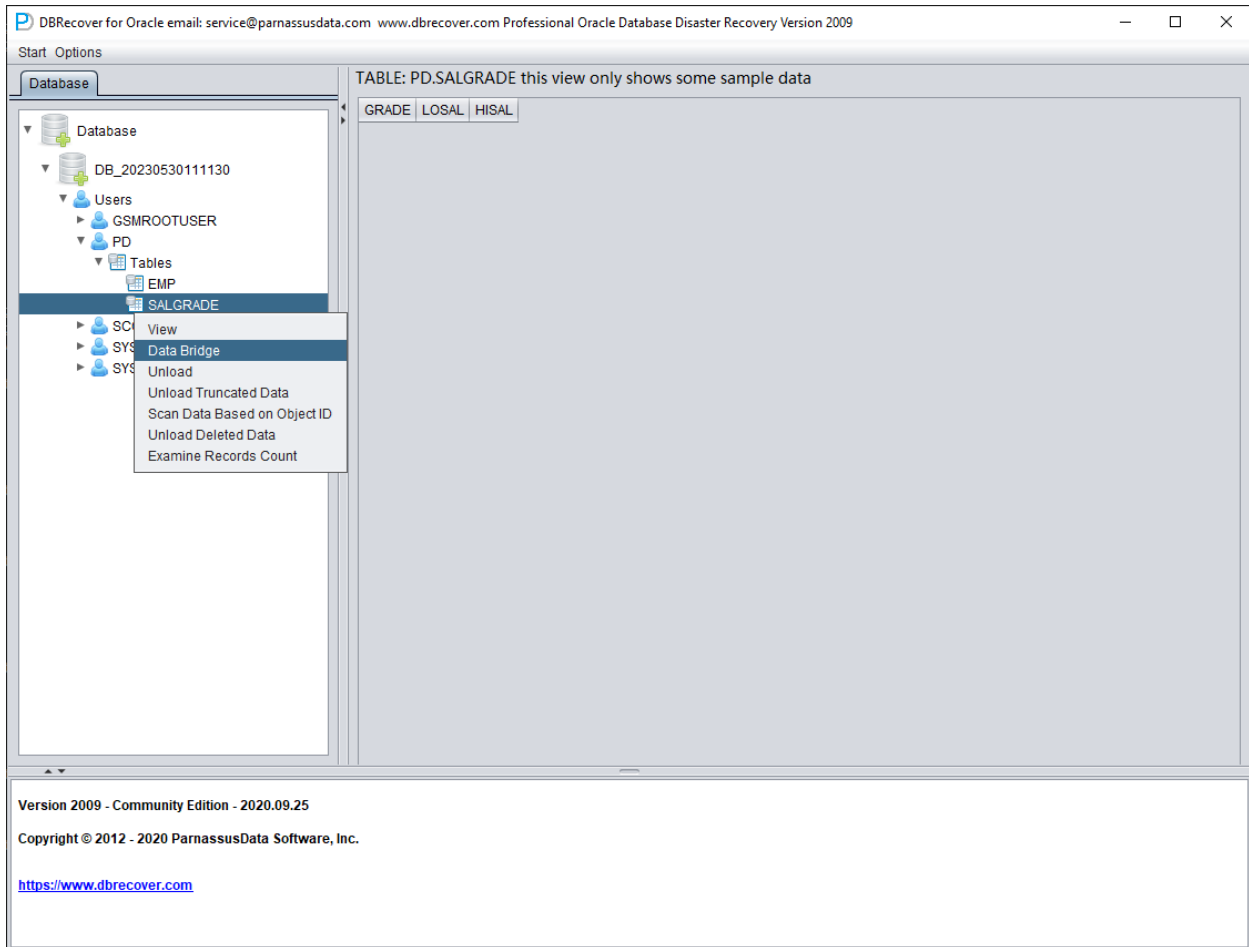
```
SQL> select user# from sys.user$ where name='PD';  
USER#  
-----  
106  
  
SQL> select obj#,dataobj# from sys.obj$ as of timestamp systimestamp -1/24 where name='SALGRADE' and owner#=106;  
OBJ# DATAOBJ#  
-----  
76112 76112
```

L'image ci-dessus montre l'utilisation d'une requête flashback pour obtenir le DATAOBJ# original, c'est-à-dire le DATA_OBJECT_ID.

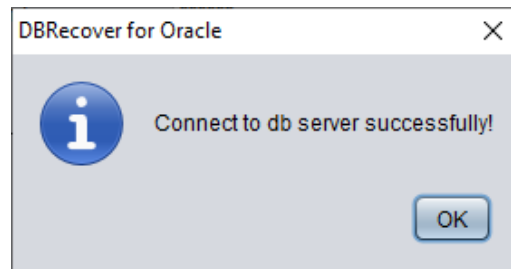
Ensuite, nous devons utiliser la fonctionnalité Data Bridge pour insérer les données à récupérer dans la base de données cible. Lors de l'utilisation de Data Bridge pour récupérer des données tronquées, veuillez noter que lorsque vous récupérez des données tronquées de la base de données source, si vous utilisez l'option Data Bridge pour transférer les données vers votre base de données source (si le transfert des données n'est pas vers la base de données source, ce problème ne se pose pas), vous devez vous assurer que l'emplacement de la nouvelle table dans laquelle Data Bridge insère les données n'est pas l'espace de table dans lequel les données tronquées de la base de données source se trouvent, et évitez d'insérer dans la table source, sinon vous risquez de récupérer les données tronquées en même temps que les données que nous devons récupérer sont recouvertes par de nouvelles données, ce qui pourrait rendre les données de ce scénario de récupération complètement irrécupérables. Par conséquent, veuillez noter que lorsque vous utilisez Data Bridge pour récupérer des données dans la base de données source, vous ne devez jamais utiliser l'espace de table où se trouvent les données à récupérer dans Data Bridge !!!!!!!

Donc, nous créons d'abord un nouvel espace de table pour stocker la table de données récupérée ici :

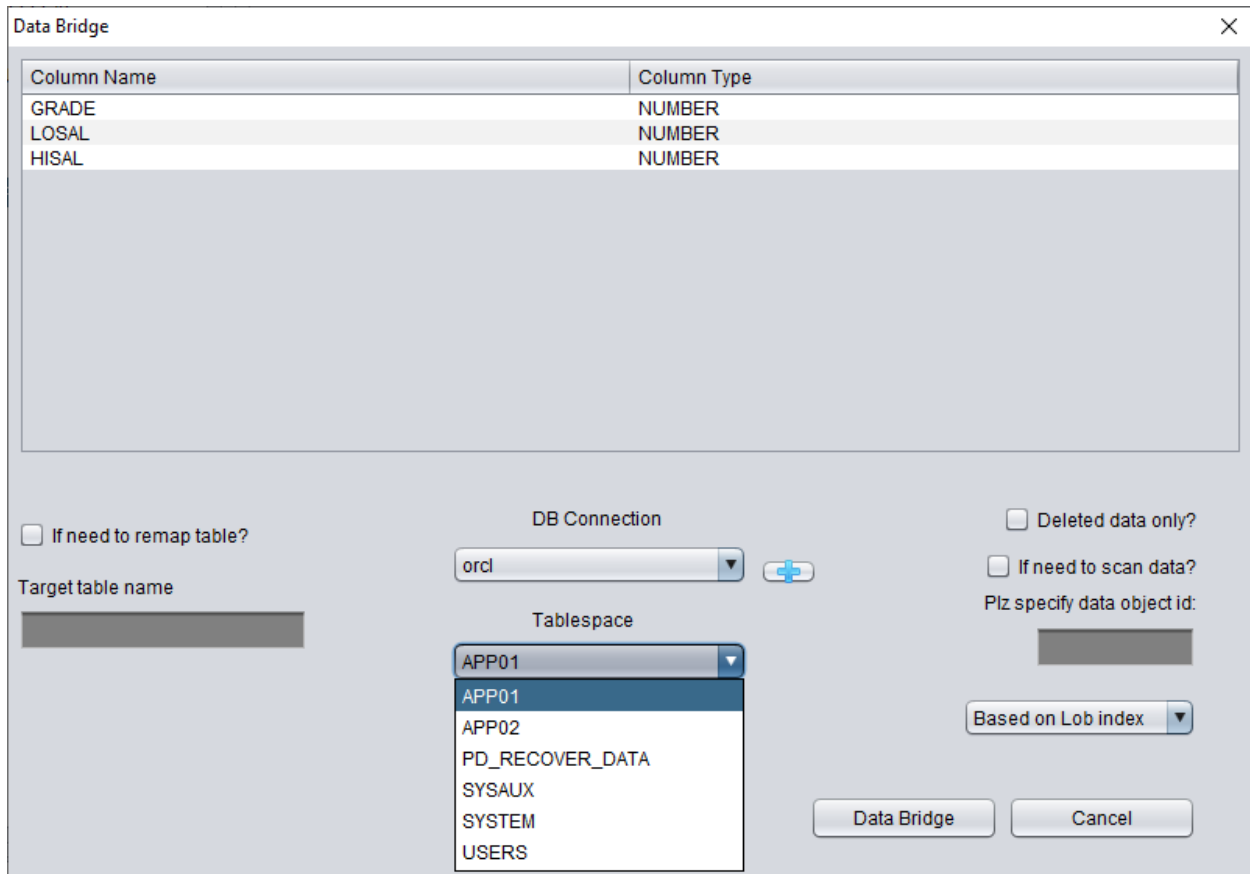
```
SQL> create tablespace pd_recover_data datafile size 600M;  
Tablespace created.
```



Créez les informations de connexion nécessaires, notez que l'utilisateur de la base de données doit avoir les permissions nécessaires (il est recommandé d'accorder le rôle DBA).



Après un test réussi, cliquez sur **SAVE** pour sauvegarder.



L'image ci-dessus sélectionne l'espace de table pour stocker la table de données tronquée récupérée.

Column Name	Column Type
GRADE	NUMBER
LOSAL	NUMBER
HISAL	NUMBER

If need to remap table?
 Target table name

DB Connection

Tablespace

Deleted data only?
 If need to scan data?
 Plz specify data object id:

Ici, nous devons cocher "if need to scan data" et entrer l'original DATA_OBJECT_ID que nous avons obtenu précédemment, de sorte que DBRECOVER scannera explicitement les données correspondant à cet ID pour nous.

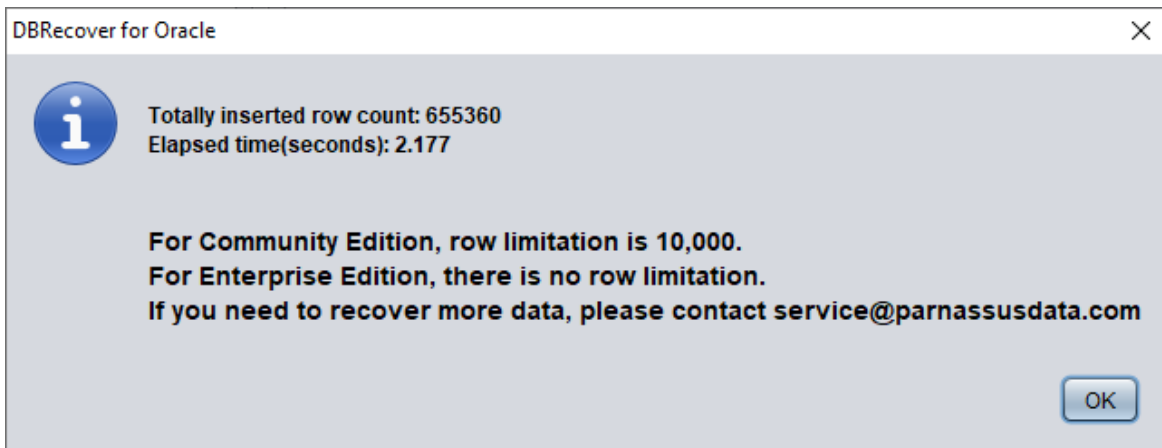
En même temps, nous devons cocher "if need to remap table", et entrer un nouveau nom de table, afin de permettre l'insertion des données dans la nouvelle table (sous le nouvel espace de table), en éliminant toute possibilité de recouvrement des données.

Remarque :

1. Pour les cas où le nom de la table correspondante existe déjà dans la base de données cible, DBRECOVER n'essayera pas de recréer la table, mais insérera les données à récupérer sur la base de la table existante, donc l'espace de table spécifié sera invalide

2. Pour les cas où le nom de la table correspondante n'existe pas encore dans le SCHEMA de la base de données cible, DBRECOVER essaiera de créer une table dans l'espace de table spécifié et d'y insérer les données récupérées

Après avoir terminé les étapes ci-dessus, cliquez sur le bouton Data Bridge.



Confirmez le nombre de lignes récupérées :

```
SQL> select count(*) from pd.salgrade_recover;
```

```
COUNT(*)
```

```
-----
```

```
655360
```

Le principe de base des données tronquées est que lorsque TRUNCATE se produit, ORACLE ne met à jour que l'ID de l'objet de données de la table dans le dictionnaire de données et l'en-tête du segment, mais ne modifie pas les blocs de la partie de données réelle. Comme l'ID de l'objet de données du dictionnaire de données et de l'en-tête de segment ne correspond pas à celui des

blocs de données subséquents, le processus de service ORACLE ne lira pas les données qui ont été tronquées mais qui n'ont pas été réellement recouvertes lors de la lecture des données de la table entière. Par conséquent, DBRECOVER peut récupérer les données à partir des zones de disques de données (Data Extent) qui n'ont pas été modifiées ou recouvertes.

Scénario de récupération 6 : Récupération après une suppression accidentelle d'une table

Un développeur d'applications de la société D a DROP une table d'application centrale dans le système sans aucune sauvegarde, à ce moment, DBRECOVER peut être utilisé en premier pour récupérer la majorité des données de la table DROP. Après 10g, la fonctionnalité de corbeille recyclebin est disponible, vous pouvez d'abord déterminer si la table supprimée se trouve dans la corbeille en consultant la vue DBA_RECYCLEBINS, si elle y est, préférez utiliser flashback to before drop à partir de la corbeille, si elle n'est pas dans la corbeille, utilisez DBRECOVER pour la récupération dès que possible.

Semblable à la récupération TRUNCATE, la récupération d'une table DROP nécessite de déterminer l'original DATA_OBJECT_ID.

Le processus de récupération est brièvement comme suit :

1. Tout d'abord, définir l'espace de table dans lequel la table supprimée se trouve en mode lecture seule (ALTER TABLESPACE {TABLESPACE_NAME} READ ONLY ; ou copier tous les fichiers de données de l'espace de table entier dès que possible
2. Trouver le DATA_OBJECT_ID de la table supprimée en consultant le dictionnaire de données ou LOGMINER
3. Démarrer DBRECOVER en mode NON-DICT non-dictionnaire, et ajouter tous les fichiers de données de l'espace de table où se trouve la table supprimée, puis SCAN DATABASE + SCAN TABLE from Extent MAP
4. Localiser la table correspondante dans l'arbre d'objets déplié par DATA_OBJECT_ID, et l'insérer de nouveau dans la base de données source en mode Data Bridge

Vous pouvez obtenir l'approximation de DATA_OBJECT_ID à travers logminer, le script pour utiliser LOGMINER est approximativement comme suit :

```

EXECUTE DBMS_LOGMNR.ADD_LOGFILE( LOGFILENAME => '/oracle/logs/log1.f', OPTIONS => DBMS_LOGMNR.NEW);
EXECUTE DBMS_LOGMNR.ADD_LOGFILE( LOGFILENAME => '/oracle/logs/log2.f', OPTIONS =>
DBMS_LOGMNR.ADDFILE);
Execute
DBMS_LOGMNR.START_LOGMNR(DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG+DBMS_LOGMNR.COMMITTED_DATA_ONLY);
SELECT * FROM V$LOGMNR_CONTENTS ;
EXECUTE DBMS_LOGMNR.END_LOGMNR;

```

Vous pouvez également essayer d'obtenir DATA_OBJECT_ID en fouillant les données AWR :

```

Select * from
(select object_name,object# from DBA_HIST_SQL_PLAN
UNION select object_name,object# from GV$SQL_PLAN) V1 where V1.OBJECT# IS
NOT NULL minus select name,obj# from sys.obj$;

select obj#,dataobj#, object_name from WRH$_SEG_STAT_OBJ where object_name
not in (select name from sys.obj$) order by object_name desc;

SELECT tab1.SQL_ID,
current_obj#,
tab2.sql_text
FROM DBA_HIST_ACTIVE_SESS_HISTORY tab1,
dba_hist_sqltext tab2
WHERE tab1.current_obj# NOT IN
(SELECT obj# FROM sys.obj$
)
AND current_obj#!=-1
AND tab1.sql_id =tab2.sql_id(+);

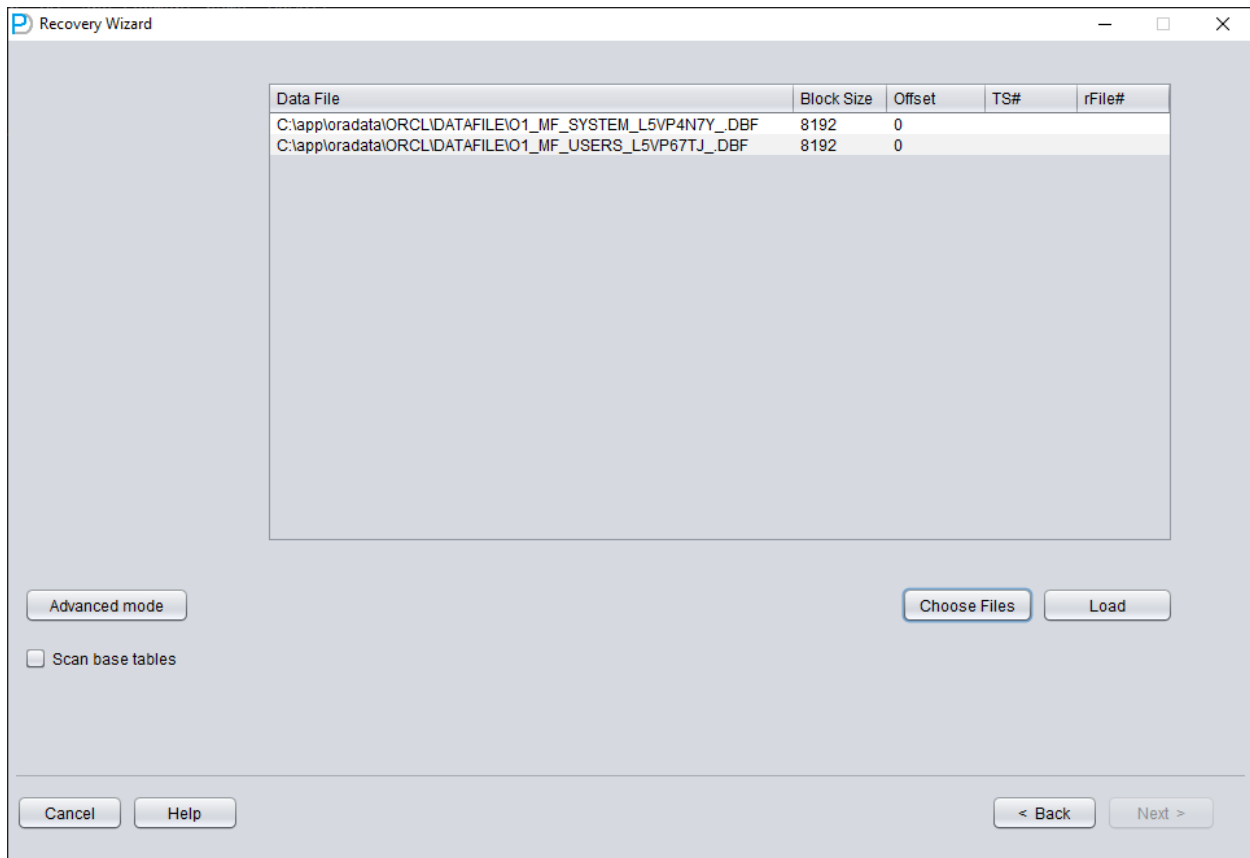
```

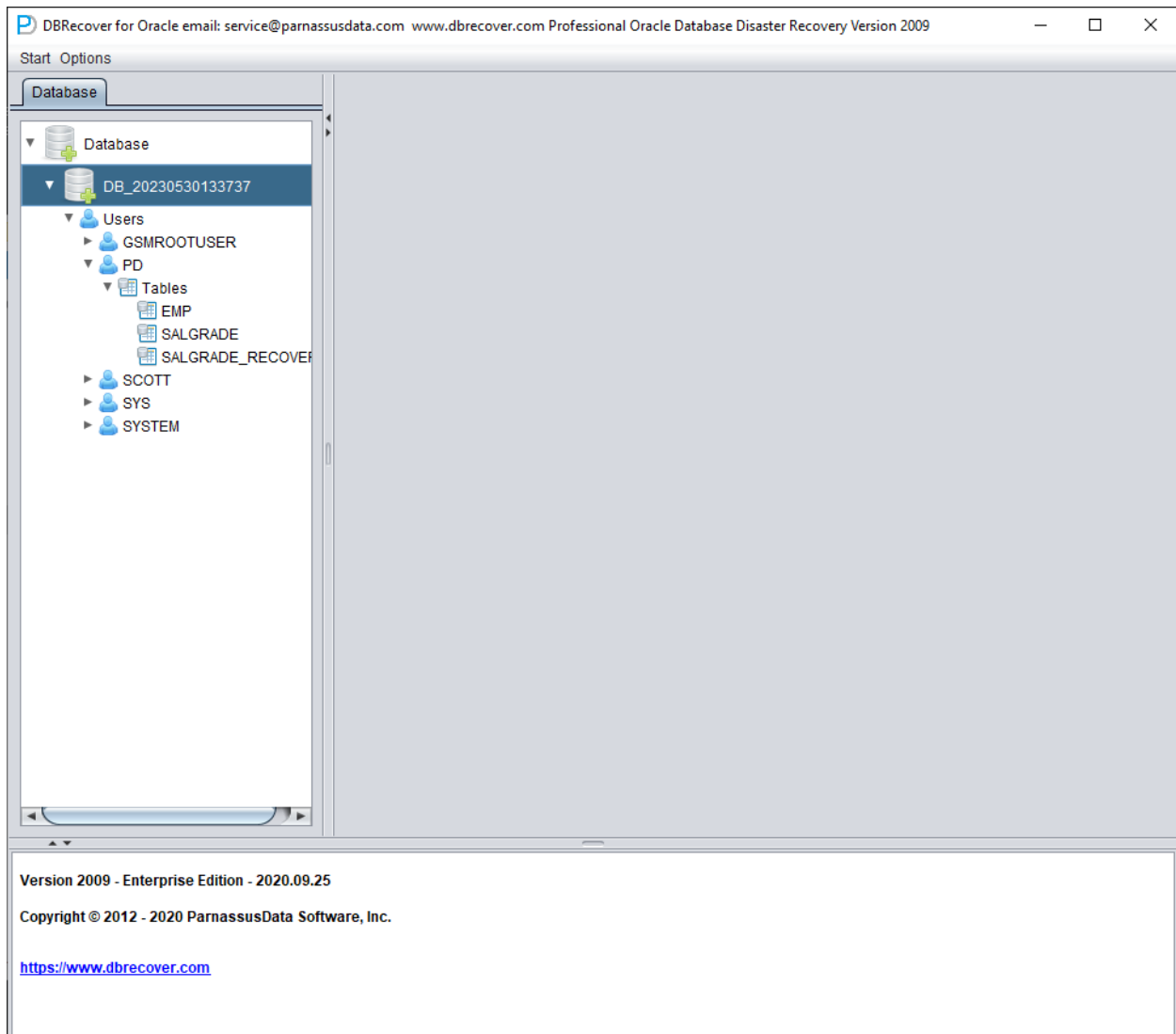
//Les trois requêtes ci-dessus comparent les données AWR avec la table de base du dictionnaire OBJ\$, afin de découvrir la table supprimée

Laissez-nous faire une démonstration en temps réel :

```
SQL> create table dropit as select * from dba_objects;
Table created.
SQL> select count(*) from pd.dropit;
COUNT(*)
-----
73095
SQL> select tablespace_name from dba_segments where owner='PD' and segment_name='DROPIT';
TABLESPACE_NAME
-----
USERS
SQL> select object_id ,data_object_id from dba_objects where owner='PD' and object_name='DROPIT';
OBJECT_ID DATA_OBJECT_ID
-----
76116 76116
SQL> drop table dropit;
Table dropped.
SQL> alter system checkpoint;
System altered.
```

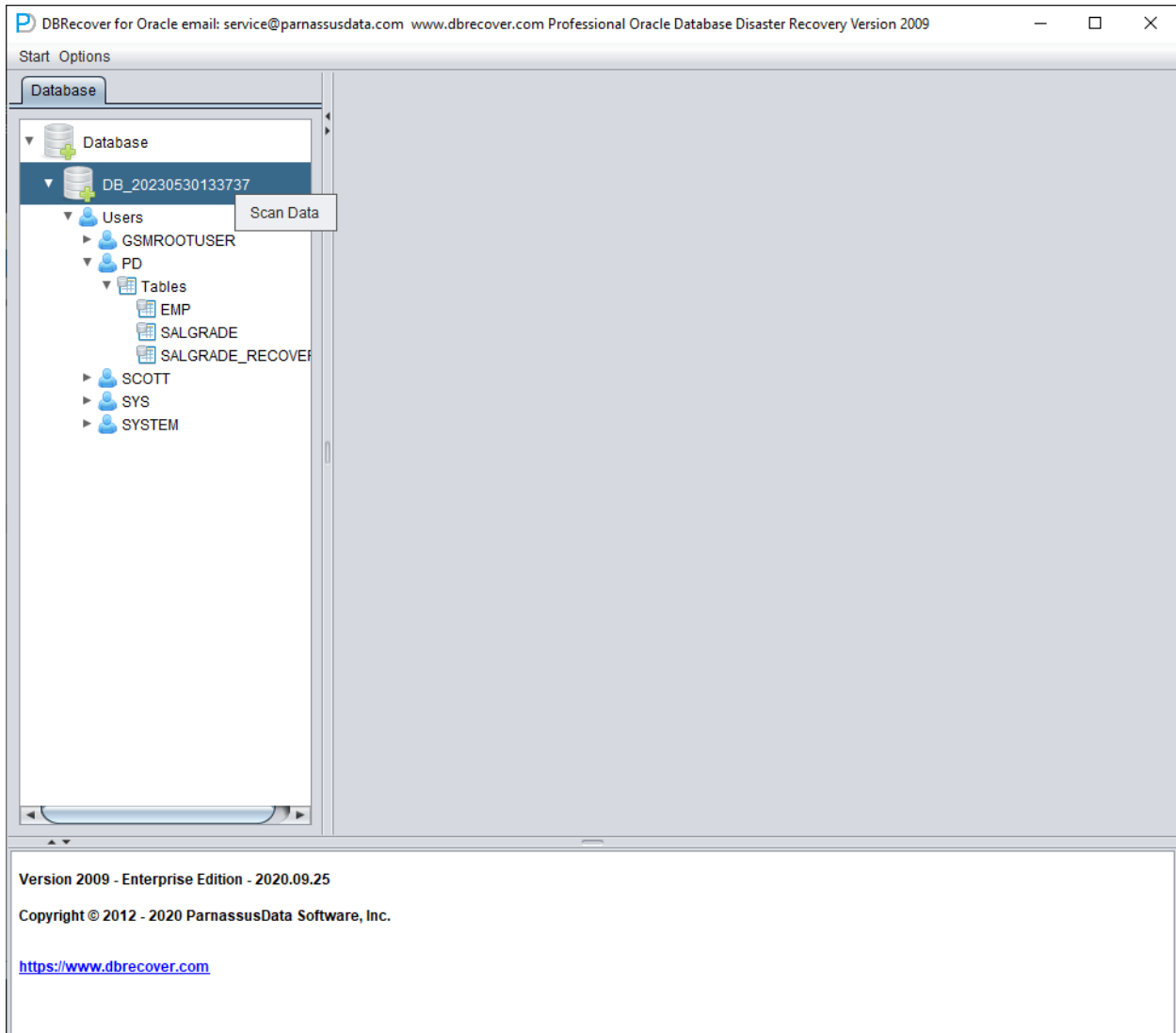
Nous démarrons DBRECOVER en mode dictionnaire (DICTIONARY-MODE), ici nous n'avons besoin d'ajouter que SYSTEM01.DBF et l'espace de table USERS où se trouve la table :

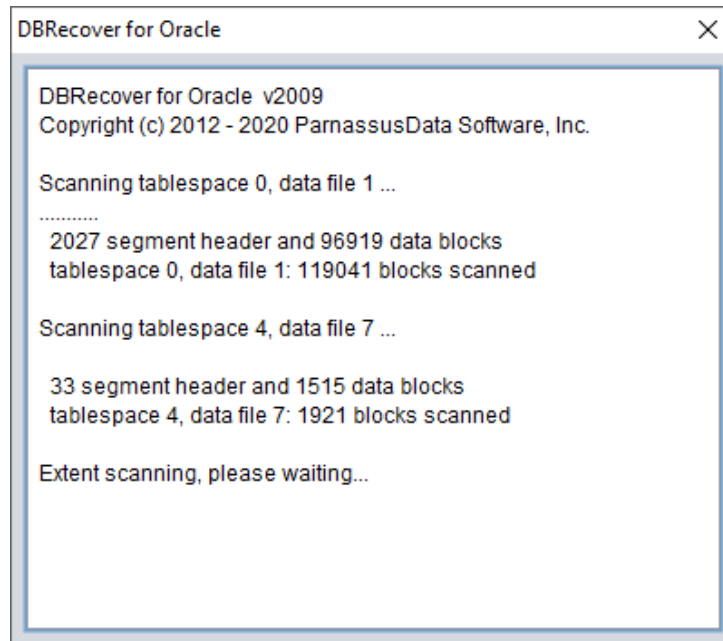


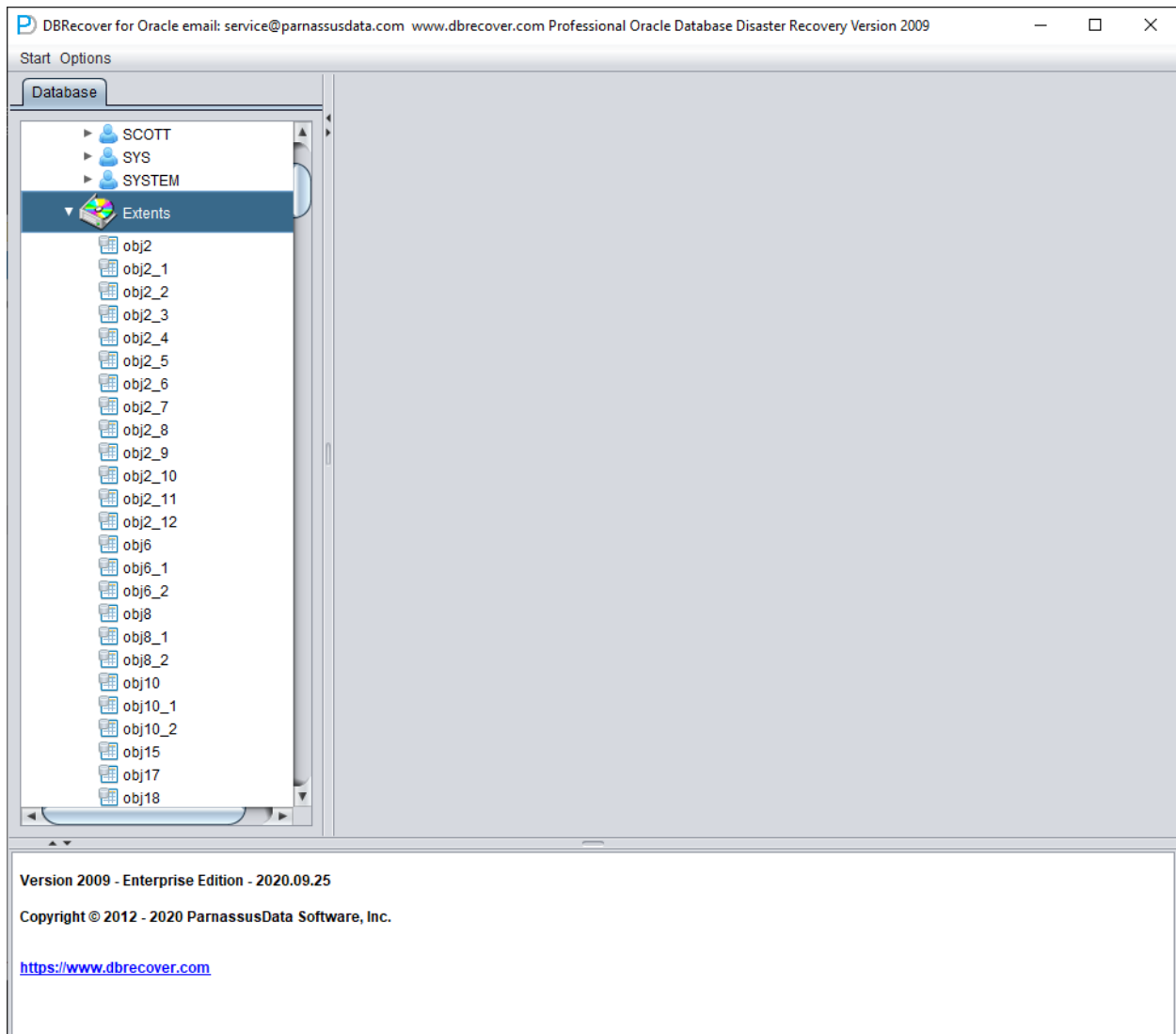


Une fois le chargement terminé, nous pouvons voir que la table que nous voulons récupérer n'existe pas sous le SCHEMA PD, ce qui est normal.

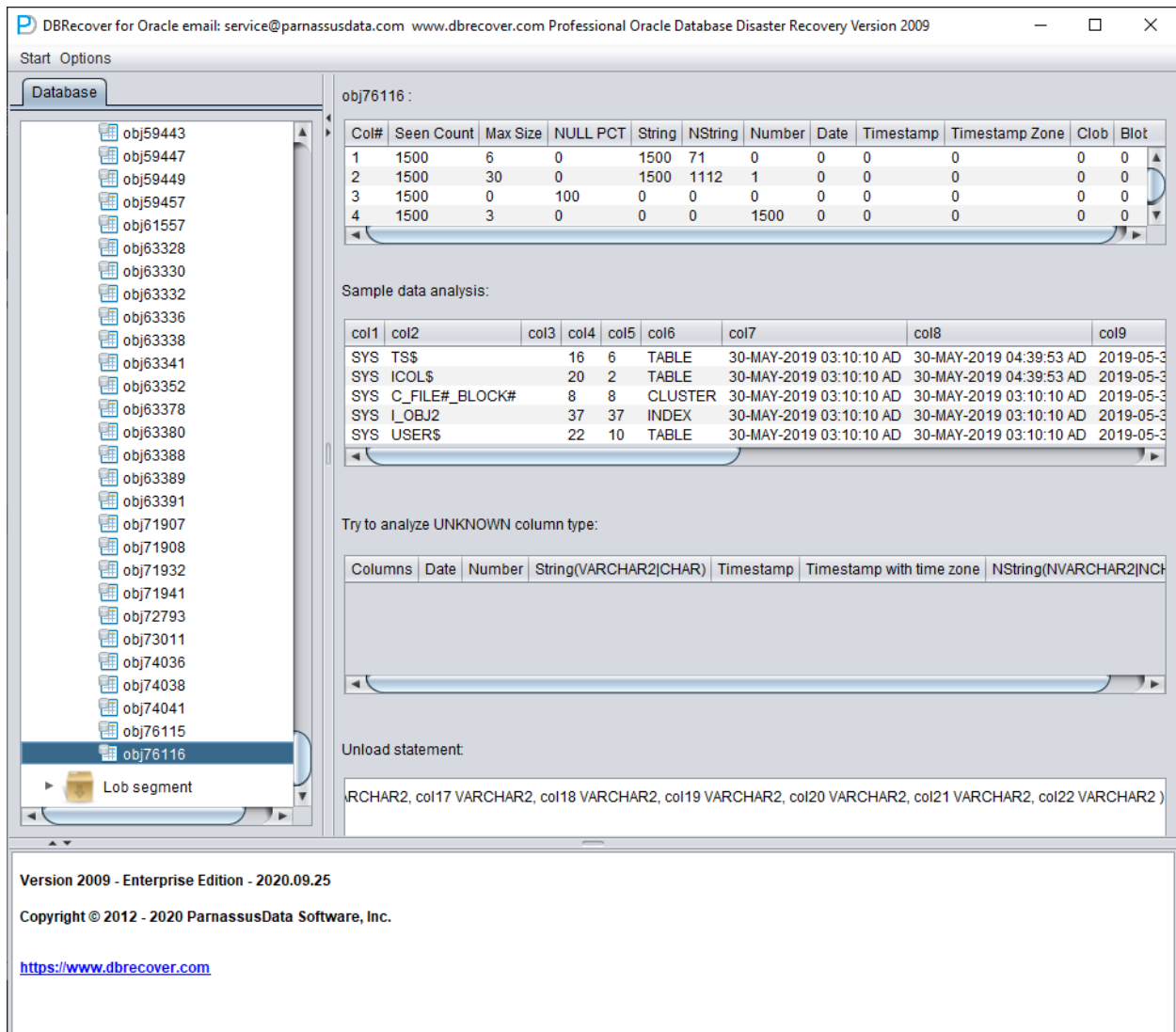
Sélectionnez le nœud de la base de données, clic droit sur SCAN Data







Ensuite, un nœud EXTENTS apparaîtra, cherchez le nœud OBJ76116 :



Ensuite, nous pouvons utiliser la fonctionnalité Data Bridge pour l'insérer de nouveau dans la base de données source.

Scénario de récupération 7 : Récupération après une suppression accidentelle de TABLESPACE

Un employé de la société D avait besoin de supprimer un espace de table inutile, c'est-à-dire une opération `DROP TABLESPACE INCLUDING CONTENTS`, mais après l'opération `DROP TABLESPACE`, le département de développement a signalé que les données d'un SCHEMA sur le TABLESPACE supprimé étaient en fait utiles et importantes, mais maintenant l'espace de table a été supprimé et il n'y a pas de sauvegarde.

Dans ce cas, vous pouvez utiliser le mode `NON-DICTIONARY` de `DBRECOVER` pour extraire les données de tous les fichiers de données correspondants du TABLESPACE supprimé. Cette méthode peut récupérer la plupart des données, mais comme il s'agit d'un mode non-dictionnaire, il est nécessaire de correspondre une à une les tables récupérées avec les tables de données d'application. A ce moment, l'intervention des développeurs d'application est généralement nécessaire, en identifiant manuellement à quelle table appartiennent les données. Comme l'opération `DROP TABLESPACE` a modifié le dictionnaire de données et supprimé les objets sur l'espace de table correspondant dans `OBJ$`, il n'est pas possible d'obtenir la correspondance entre `DATA_OBJECT_ID` et `OBJECT_NAME` à partir de `OBJ$`. A ce moment, nous pouvons utiliser la méthode présentée dans le scénario `DROP TABLE` pour obtenir autant que possible la correspondance entre `DATA_OBJECT_ID` et `OBJECT_NAME`.

Le processus général est comme suit :

Si les fichiers de données sont également supprimés physiquement lors de la suppression de TABLESPACE, il faut d'abord récupérer les fichiers de données. Vous pouvez essayer d'utiliser un logiciel de récupération au niveau du système de fichiers, ou utiliser le logiciel `PRMSCAN` pour scanner et réorganiser les fichiers de données au niveau des blocs de données ORACLE. `PRMSCAN` est un outil de fusion et de balayage des fragments de blocs de données ORACLE, qui est adapté aux scénarios suivants :

1. Suppression manuelle accidentelle de fichiers de données sur n'importe quel système de fichiers (NTFS, FAT, EXT, UFS, JFS, etc.) ou sur ASM
2. Dommages au système de fichiers, entraînant une taille de fichier de données de 0 byte, c'est-à-dire une réinitialisation du fichier de données
3. Dommages au système de fichiers, rendant impossible le `MOUNT` du système de fichiers
4. Dommages aux métadonnées de stockage ASM, rendant impossible le `MOUNT` du `diskgroup`

5. Destruction ou perte physique du LV ou PV dans le système de fichiers ou ASM
6. Dans tous les scénarios ci-dessus, prmscan peut être utilisé pour scanner directement les blocs oracles résiduels non recouverts dans le PV ou LV correspondant du système de fichiers ou ASM, afin de réaliser la fusion et la réorganisation de ces blocs de données oracles, dans le but de récupérer les données.

PRMSCAN est développé en JAVA, et peut fonctionner sur tous les systèmes d'exploitation qui supportent JDK 1.6 et plus, y compris Windows, Linux, Solaris, AIX, HP-UX.

Actuellement, ce produit n'est pas disponible à la vente au détail, mais vous pouvez nous contacter pour obtenir des services de récupération.

Par exemple, dans l'exemple ci-dessous, /dev/sdb1 est une partition du système de fichiers ext4, mais en raison des dommages au système de fichiers ext4, SDB1 ne peut pas être MOUNT. Cependant, ce système de fichiers contient les fichiers de données d'une base de données oracle, et si le système de fichiers ne peut pas être MOUNT, la base de données oracle ne pourra pas non plus être utilisée.

Ici, nous utilisons la fonction de balayage et de fusion de prmscan pour réorganiser tous les fichiers de données directement à partir du système de fichiers endommagé.

Pour scanner l'ensemble du disque :

```
[oracle@dbdao01 ~]$ java -jar PRMScan.jar -scan /dev/sdb1 -guess 8k
```

L'option -scan représente le balayage de l'appareil /dev/sdb1 et spécifie que le blocksize Oracle est de 8k

```
[oracle@dbdao01 ~]$ java -jar PRMScan.jar -outputsh ./8kfull.txt
```

-outputsh représente l'écriture d'un fichier SHELL qui peut fusionner les informations déjà balayées, à savoir ici 8kfull.txt

```
[oracle@dbdao01 ~]$ sh 8kfull.txt
```

En exécutant 8kfull.txt, vous pouvez générer tous les fichiers de données nécessaires à la fusion dans le répertoire actuel

```
[oracle@dbdao01 ~]$ ls -ll PD*
```

```
rw-r--r-- 1 oracle oinstall 295428096 Jul 28 00:37 PD_DBF1.dbf
```

```
rw-r--r-- 1 oracle oinstall 83427328 Jul 28 00:37 PD_DBF2.dbf
```

```
rw-r--r-- 1 oracle oinstall 220266496 Jul 28 00:37 PD_DBF3.dbf
```

```
rw-r--r-- 1 oracle oinstall 1324482560 Jul 28 00:38 PD_DBF4.dbf
```

Si les fichiers de données n'ont pas été supprimés physiquement, vous pouvez les ajouter directement à DBRECOVER en mode non-dictionnaire (NON-DICTIONARY MODE) et scanner les données qu'ils contiennent.

Les étapes suivantes peuvent être consultées dans l'opération DROP TABLE ci-dessus, la différence étant que l'objet de récupération de DROP TABLESPACE sera de nombreuses tables.