أ/عبدالقادر البعداني abdalkader_2015@hotmail.com

محاضرات إدارة قواعد البيانات(DBA) أوراكل ١٠جي

اليمن – اب

جامعة الجزيرة

٢٠١٠-٢٠١١م

أ/عبدالقادر البعداني abdalkader_2015@hotmail.com

إدارة قواعد البيانات

ما هي اوراكل :

هي شركة مختصة في مجال قواعد البيانات تقدم مجموعة هائلة من المنتجات البرمجية في هذا المجال و تشمل على ثلاث فئات أساسية : نظام إدارة قواعد بيانات ، نظم تطوير تطبيقات لاستخلاص البيانات من نظام الإدارة ، نظم جاهزة تضمن تلبية حاجات السوق البرمجي ..

تضم الأوراكل مجالاً واسعاً من المنتجات :

1- Oracle Database Server : و هو نظام إدارة قواعد البيانات حيث يتم تخزين البيانات و إدارتها و تنظيم المستخدمين ... إلخ و هو محور بحثنا في هذه الدورة .

2- Oracle Developer Tools و هي أدوات تمكنك من تصميم تطبيقات تعتمد على قاعدة بيانات أوراكل و تضم الأمر الشائعة التالية :

1- Oracle Forms Builder و هو الأداة الأساسية لتصميم واجهات برامج كالتي نراها و لكن موجهة نحو تطبيقات مستعرض الإنترنت .

2- Oracle Reports و هي أداة رائعة لإنتاج التقارير و هي تشابه كريستال ريبورت التي إعتاد عليها مطورو منصة .net.

3- Oracle Designer و هو أداة رائعة تتيح لك تصميم قاعدة البيانات بأسلوب منهجي يعتمد على رسم كائنات Entity .

4- JAVA developer : و هي أداة تطوير برامج الجافا التي تعتمد على قاعدة البيانات أوراكل و هي مشابه لبيئة JBUILDER التي إعتاد عليها مطورو الجافا في بناء تطبيقاتهم ..

5- ORACLE provider for .net Developer : و هي مجموعة من الأدوات الموجهة لمطوري .net الذين يرغبون ببناء تطبيقاتهم اعتمادا على قاعدة بيانات أوراكل ..

3- Oracle application Server : و هو المنصة التي نقوم بعملية النشر لتطبيقات Oracle forms و Oracle reports عليها بحيث تصبح متاحة للجميع ، و هي تشبه مخدم الويب IIS لنشر صفحات ASP أو Apache لصفحات .. php

4- Oracle Collaboration Suite : و هي مجموعة من الخدمات المختصة بالاتصالات الصوتية أو الإلكترونية.

Database Administration أنواع مستخدمي قاعدة البيانات

تتنوع مهام ووظائف المعنيين بإدارة قواعد بيانات الأوراكل طبقا لحجم العمل وحجم قاعدة البيانات وأهمية قاعدة البيانات ، وبناء علي تلك العوامل يمكن تخصيص أو دمج المهام المتعلقة بإدارة وتشغيل قاعدة البيانات . ولكن يمكننا أن نذكر التخصصات الموصي بها من قبل شركة أوراكل في توزيع المهام حسب الوظائف التالية :

- مدير قاعدة البيانات Database Administrator
- مسئول أمن البيانات Security Officer
- مدير الشبكة Network Administrator
- مطور التطبيقات Application Developer
- مدير التطبيقات Application Manager
- مستخدم قاعدة البيانات Database User

مدير قاعدة البيانات Database Administrator

مدير قاعدة البيانات هو الشخص المعني بعدد من المهام الأساسية واللازمة لاستمرار عمل قاعدة البيانات بشكل سليم ويمكن تلخيص مهامه بآلاتي :

- **تركيب وإعداد خادم الأوراكل والتطبيقات المتعلقة بالأوراكل .**
- **التحديث والتطوير لقاعدة البيانات .**
- **تحديد الموارد المطلوبة مثل مساحات وسائط التخزين وحجم الذاكرة ، والتوصية بتحديثها أو تعديلها بناء علي الأداء .**
- **بناء الهياكل الرئيسية للبيانات مثل الجداول الفهارس بالتواصل مع طاقم التطوير .**
- **تعديل وتحديث قاعدة البيانات بناء علي متطلبات المبرمجين .**
- **مراقبة أداء قاعدة البيانات ومعالجة مشاكلها .**
- **مراقبة المستخدمين لقاعدة البيانات والتأكد من كفاءة الصلاحيات .**
- **التخطيط والتنفيذ لعمليتي النسخ و الاسترجاع لقاعدة البيانات .**
- **تنظيم وإدارة عملية أرشفة قاعدة البيانات .**
- **التواصل وتقديم الدعم الفني لكل الجهات التي تتعامل مع قاعدة البيانات وفي مجال اختصاصه .**

مسئول أمن البيانات Security Officer

في بعض الحالات تكون قاعدة البيانات موزعة بشكل هرمي ، وذلك بناء علي مستويات صلاحية مركبة وفي هذه الحالة يتم تخصيص مسئول أمن بيانات أو اكثر لإدارة توزيع الصلاحيات ، وفي هذه الحالة تكون إدارة الصلاحيات خارج نطاق عمل مدير قاعدة البيانات ويكون مسئول أمن البيانات معني بالاتي:

- **توزيع الصلاحيات والتأكد من مطابقة الصلاحيات لواقع العمل.**
- **مراقبة المستخدمين والتأكد من صحة إعداد الصلاحيات الممنوحة لهم.**

مدير الشبكة Network Administrator

بعض قواعد بيانات الأوراكل يتم فيها تخصيص مدير واحد للشبكة أو أكثر بغرض إدارة اتصالات الأوراكل.

مطور التطبيقات Application Developer

يقوم مطور البيانات بتطوير تطبيقات الأوراكل المبنية علي إجراءات العمل وتكون مهامه كما يلي :

- **تصميم وبناء التطبيقات المرتبطة بقاعدة البيانات .**
- **تحديد هياكل البيانات التي ستتعامل معها التطبيقات مناقشة الموارد المطلوبة مع مدير قاعدة البيانات واللازمة لعمل التطبيقات مثل مساحة التخزين وغيرها من الموارد .**

مدير التطبيقات Application Administrator

في بعض الأنظمة المعقدة والكبيرة قد يحتاج كل تطبيق أو مجموعة من التطبيقات إلي مدير خاص به ، يشرف علي إدارة تطويره والتأكد من عمله بشكل سليم .

مستخدم قاعدة البيانات Database User

تحتاج بعض قواعد البيانات إلي التعامل مباشرة مع البيانات ، ولهذا يتم تخصيص وظيفة مستخدم قاعدة بيانات بحيث تكون مهمته هي :

- **إدخال وتعديل وحذف البيانات مباشرة باستخدام عباراتSQL .**
- **كتابة الاستعلامات وإصدار التقارير المطلوبة يدويا .**
- **تحميل بعض البيانات من المصادر الخارجية .**
- **استخراج البيانات لتتوافق مع نظم خارجية .**

إدارة المستخدمين

ما هي الحاجة إلى وجود مستخدمين ؟

تكمن الحاجة في حتمية وجود خصوصية للبيانات بحيث يستطيع كل شخص أن يحتفظ ببياناته و قدرته على تعديلها و منح الحق لغيره بالاطلاع عليها أو إجراء العمليات المختلفة عليها .

قبل البدء بعملية إنشاء المستخدمين يجب أن أوضح المصطلحات التالية - :

Schema المخطط :و هي مجموعة objects التي يملكها المستخدم أي الجداول PL/SQL– index –

CODE .... –

Table Spaceالفضاء الجدولي :منطقياً هو المكان المخصص لكل مستخدم لكي يقوم بتخزين بياناته عليه ، فيزيائياً هو مجموعة من الملفات التي تخزن البيانات ندعوها . data file

Privilegesالصلاحية :و تعني إمكانية قيام مستخدم معين بعملية ما على البيانات أو إنشاء بنى جديدة أو حذف بنى موجودة ....إلخ .

Quotas : و هي المساحة المخصصة للمستخدم للتخزين على فضاء tablespace معين .

أو هى الحصة المتاحة للمستخدم لإنشاء الكائنات فى الTablespaces وبدونها يكون المستخدم عاجز عن إنشاء اى كائن .

لذا يمكن لمدير قاعدة البيانات منع المستخدم من إنشاء أى كائن عن طريق منحه حصة Quota تساوى صفر فى الTablespace .

كذلك يستطيع مدير قاعدة البيانات منح المستخدم حصة Quota غير محدودة فى الTablespace عن طريق منحه Unlimited .

يمكن منح الحصة للمستخدم اثناء الإنشاء او بعد ذلك .

تعليمة إنشاء : user

```
CREATE USER user
IDENTIFIED {BY password }
[ DEFAULT TABLESPACE tablespace ]
[ TEMPORARY TABLESPACE tablespace ]
[ QUOTA {integer [K | M ] | UNLIMITED } ON
tablespace]
[ PASSWORD EXPIRE ]
[ ACCOUNT { LOCK | UNLOCK }]
[ PROFILE { profile | DEFAULT }]
```

1-إنشاء : Database User
الإنشاء المثال التالي:

```
create user jazeera
identified by jazeera_pas;
```

إذا أردنا أن نتوسع أكثر فلنجعل لهذا المستخدم حصة على Users Tablespace و هو فضاء جدولي افتراضي يتم إنشاءه عند تنصيب : Oracle

```
create user jazeera identified by jazeera_pas
default tablespace  users
quota 50 m on users;
```

يمكن منح الحصة للمستخدم اثناء الإنشاء او بعد ذلك .

فلنفترض الان اننا نريد منح حصة تساوى صفر للمستخدم jazeera فى Users Tablespace .

```
ALTER USER jazeera QUOTA 0M ON USERS;
```

ولنفترض الان اننا نريد منحه حصه غير محدودة فى Users Tablespace .

```
ALTER  USER  jazeera QUOTA UNLIMITED ON USERS;
```

لو أننا طلبنا من المستخدم أن يقوم بعملية تغيير قسرية لكلمة السر الخاصة به عند أول دخول ؟؟

```
create user jazeera
identified by jazeera _pass
default tablespace  users
        quota 50m on users
password Expire;
```

إجراء التعديلات على المستخدمين الذين أنشأناهم ؟ هل نستطيع تغيير كلمة المرور أو Quota أو الفضاء الجدولي ؟؟؟

و ذلك عن طريق التعليمة :

```
ALTER USER user
[ DEFAULT TABLESPACE tablespace]
[ TEMPORARY TABLESPACE tablespace]
[ QUOTA {integer [K | M] | UNLIMITED } ON
tablespace
...]
```

تغيير كلمة المرور للمستخدم الذي أنشأناه ؟؟

```
SQL> alter user jazeera
  2  identified by jazeera_pass;

User altered.
```

ملاحظة : يمكن حتى للمستخدم jazeera نفسه أن ينفذ هذه التعليمة لتغيير كلمة السر الخاصة به .

تغيير المساحة المخصصة للمستخدم للكتابة على الفضاء الجدولي USERS ؟؟؟

```
SQL> alter user jazeera
  2  quota 50m  on users;
User altered
```

إغلاق حساب المستخدم :

```
SQL> alter user jazeera account lock;
```

فتح حساب المستخدم :

```
SQL> alter user jazeera account unlock;
```

و الآن لنسجل الدخول بالمستخدم الذي أنشأناه :

```
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create user jazeera
  2  identified by jazeera_pass;

User created.

SQL> connect jazeera / jazeera_pass
ERROR:
ORA-01045: user jazeera lacks CREATE SESSION privilege; logon denied


Warning: You are no longer connected to ORACLE.
SQL>
```

ما الذي حدث ؟؟؟؟

تحدثنا في البداية بضرورة وجود الصلاحيات التي تحكم عمل المستخدم حتى في تسجيل دخوله . لذلك نحتاج إلى منح المستخدم jazeera القدرة على تسجيل الدخول و ذلك بالطريقة:

```
SQL> connect system/pass
Connected.
SQL> grant create session to jazeera;
Grant succeeded.
SQL> connect jazeera / jazeera _pass;
Connected.
```

٢- الصلاحيات (Privilege)

بكلمتين : الصلاحية هي قدرة المستخدم على فعل شيء ما .
و تعتبر أوراكل من أقوى نظم الإدارة التي تحوي على صلاحيات متنوعة تكاد تشمل كل شيء حتى الصلاحيات الصغيرة و هي تعتبر نقطة قوة في صالح أوراكل .
تقسم الصلاحيات إلى نوعين :
-1على مستوى النظام : System Level تسمح بتنفيذ عمليات على القاعدة بشكل عام كإنشاء الجداول و حذفها ....إلخ .
-2على مستوى الغرض : Object تسمح بتنفيذ عمليات على غرض محدد . مثلاً : حذف سجلات من الجدول emp .
يتم منح الصلاحية إلى المستخدم عن طريق :
-1مباشر بمنح الصلاحية و تسميته .
-2عن طريق Role مجموعة صلاحيات سنتعرف إليها لاحقاً .
-3عن طريق اسناد الصلاحية لجميع المستخدمين . public
و لنبدأ بـ

1- System Privilege :-

يوجد حوالي ١٠٠ صلاحية مختلفة ضمن هذا المستوى تتضمن العمليات على مستوى القاعدة للتعامل مع الأغراض المختلفة مثل الجداول و الفهارس و المتسلسلة Sequence و الفضاءات الجدولية ..... TableSpaces

```
GRANT {system_privilege|role}
[, {system_privilege|role} ]...
TO {user|role|PUBLIC}
[, {user|role|PUBLIC} ]...
[WITH ADMIN OPTION]
```

مثال :

-ما رأيكم أن نجعل المستخدم jazeera قادراً على إنشاء جدول ؟؟

```
grant create table to jazeera;
```

يمكنك أن تمنح المستخدم أكثر من صلاحية في نفس الوقت :

مثال لنجعل المستخدم jazeera قادراً على تنفيذ تعليمة select و Update على أي جدول يريده في القاعدة :

```
grant select Any table , update any table to jazeera;
```

```
select  * from scott.emp;
```

: ما ذا تعني كلمة Any ؟ و لماذا وضعناها ؟

السبب : عندما نضع Any فهذا يعني الصلاحية المطلقة الكاملة للمستخدم على مستوى القاعدة بشكل عام .

للتوضيح :

عندما نمنح المستخدم صلاحية إنشاء جدول create table فإنه يستطيع فيما بعد حذف الجدول الذي أنشأه فقط .

ما ذا لو أردنا إطلاق يده ليعتدي على جداول الأخرين و يقوم بالحذف ؟

```
grant drop any table to jazeera;
```

-حسناً : ماذا لو أردنا أن نجعل المستخدم jazeera قادراً على منح الصلاحية التي منحناها له لمستخدم آخر ؟

يمكننا ذلك إذا أرفقنا. with admin option

أي :

```
grant create table to jazeera with admin option ;
```

و بذلك يستطيع المستخدم jazeera منح هذه الصلاحية إلى مستخدم آخر ليكن jazeera2

ماذا لو غضبنا من المستخدم jazeera و أردنا سحب الصلاحية ؟

```
revoke drop any table from jazeera;
```

إذا أردت أن تعرف جميع الصلاحيات اكتب الاستعلام التالي (بعد تسجيل الدخول بإحد الحسابين SYSTEM أو SYS) :

```
select distinct privilege
from dba_sys_privs;
```

- الآن سأذكر الصلاحيات الشائعة مقسمة حسب نوع object الذي نتعامل معه :

الصلاحية	الفئة
INDEX	CREATE ANY INDEX
	ALTER ANY INDEX
	DROP ANY INDEX
TABLE	CREATE TABLE
	CREATE ANY TABLE
	ALTER ANY TABLE
	DROP ANY TABLE
	SELECT ANY TABLE
	UPDATE ANY TABLE
	DELETE ANY TABLE
SESSION	CREATE SESSION
	ALTER SESSION
	RESTRICTED SESSION

TABLESPACE CREATE TABLESPACE
ALTER TABLESPACE
DROP TABLESPACE
UNLIMITED TABLESPACE

-2على مستوى الغرض : Object Level

هنا الصلاحية تطبق على غرض محدد :جدول –مشهد – view متسلسلة - sequence إجرائية. procedure

هذه الصلاحيات معدودة على الأصابع

1- Alter :تطبق على جدول– متسلسلة - sequence إجرائية . procedure

2- Deleteتطبق على جدول – مشهد. view

3- Execute تطبق على : إجرائية . procedure

4- Index أي فهرسة : جدول – مشهد . view

5- Insert -4جدول –مشهد. view

6- Referencesجدول.

7- Select جدول –مشهد – view متسلسلة . sequence

8- Update جدول –مشهد . view

بشكل عام :

يتم منح صلاحية object privilege بالشكل التالي :

```
GRANT { object_privilege [(column_list)]
[, object_privilege [(column_list)] ]...
|ALL [PRIVILEGES]}
ON [schema.]object
TO {user|role|PUBLIC}[, {user|role|PUBLIC} ]...
  [WITH GRANT OPTION]
```

مثال : لنجعل مستخدمنا jazeera قادراً على القراءة select من الجدول : emp

```
SQL> grant select on scott.emp to jazeera;
```

و لنجعله قادراً على تنفيذ الحزمة : dbms_output

```
SQL> grant Execute  on dbms_output to jazeera;
```

ماذا لو أردنا أن نجعل jazeera قادراً على منح الصلاحية لمستخدم آخر ؟؟

```
SQL> grant select on scott.emp to jazeera with grant
option;
```

هنا يجب أن نتوقف عند مقارنة صغيرة :

ما الفرق بين grant option التي استخدمناها مع Object Level و Admin option التي استخدمناها مع

System LeveL؟

تخيلوا السيناريو التالي :

-1قمنا بمنح صلاحية ما إلى المستخدم .jazeera

-2 قام المستخدم jazeeraبمنح هذه الصلاحية إلى المستخدم . jazeera 2

-3سحبنا الصلاحية من المستخدم .jazeera

و السؤال : هل سيتأثر المستخدم jazeera 2 بعملية السحب ؟؟

و الجواب :

إذا تم استخدام admin option فإنه لن يتأثر مطلقاً و بوسعه استخدام الصلاحية .

إذا تم إستخدام grant option فإن الصلاحية ستسحب منه أيضاً .

٣- الوظائف Role

إذاً Role هي مجموع صلاحيات المختلفة التي نود إسنادها فيما بعد إلى مستخدم واحد أو أكثر أو حتى إلى مجموعة صلاحيات أخرى .

تتميز role بسهولة استخدامها المرة تلو الأخرى بحيث يتم إسنادها إلى المستخدم و سحبها منه ، كما أن سحب الصلاحية من role تؤدي إلى سحبها من المستخدم الذي أسندنا له تلك . role يمكن أن تسند أكثر من role إلى مستخدم ما ، كما يمكنك اسناد role واحدة إلى أكثر من مستخدم .

تعليمة إنشاء : role

```
SQL> create role jazeera _role;
```

بشكل عام :

```
CREATE ROLE role [NOT IDENTIFIED |
IDENTIFIED
 {BY password | GLOBALLY }]
```

-ماذا لو أردنا منح role مشروطة لمستخدم ما ؟ أي يجب عليه كتابة كلمة سر لتمكين هذه role ؟؟

```
SQL> create role jazeera _role
identified by  jazeera _pass;
```

كيف نسند الصلاحيات إلى role ؟

بالطريقة نفسها التي أسندنا فيها الصلاحيات للمستخدمين :

```
SQL> grant create table to jazeera _role;
```

الآن كيف نسند role إلى المستخدمين ؟

```
SQL> grant  jazeera _role to jazeera;
```

حتى أننا نستطيع إسناد role إلى أخرى ، فتصبح الثانية تحوي صلاحيات الأولى :

```
SQL> grant  jazeera _role1 to jazeera _rol2;
```

بشكل عام :

```
GRANT role [, role ]...
TO {user|role|PUBLIC}
[, {user|role|PUBLIC} ]...
[WITH ADMIN OPTION]
```

تمكين و إلغاء تمكين : Roles

يستطيع المستخدم jazeera أن يفعل أو يلغي عمل role معينة قام مدير القاعدة أو أي مستخدم آخر بإسنادها إليه . قد يسأل البعض :أي مجنون يلغي صلاحيات ممنوحة إليه ؟؟؟

الجواب سيدركه مطوري التطبيقات أكثر من غيرهم : في التطبيقات قد يتشارك الموظفون اسم مستخدم واحد لاستعمال التطبيق ، و لكن نريد فرض مستويات من الحماية على مستوى المستخدم الواحد ، أي قد نسمح لموظف (س (بصلاحيات أكثر من الموظف (ع) رغم أن الاثنين يقومان بتسجيل الدخول بالمستخدم .jazeera

كيف نستطيع التفعيل :

بإستخدام التعليمة : set

```
Set role jazeera _role;
```

ماذا لو أراد المستخدم أن يفعل role مشروطة بكلمة سر ؟؟

```
set role jazeera _role identified by jazeera _pass;
```

ماذا لو أراد المستخدم تفعيل جميع roles التي يملكها ماعدا role معينة ؟

```
set role all except jazeera _role;
```

ملاحظة هامة جداً : كل عمليات set التي تجريها سوف يتم التراجع عنها عند تسجيل الخروج بحيث تعود مجموعة الصلاحيات الافتراضية التي يتم إسنادها بالشكل : (توقيف الصلاحية من قبل مدير قاعدة البيانات)

```
SQL> alter user jazeera
  2  default role jazeera _role;
```

-يتم سحب الصلاحية من role بنفس الطريقة التي استعملناها مع : user

```
revoke create table from jazeera _role;
```

-حذف الصلاحية يتم بالتعليمة التالية :

```
 drop role jazeera _role;
```

ADVISOR ,AUDIT ANY ,DROP USER ,RESUMABLE ,ALTER USER ,
CREATE JOB , ANALYZE ANY , BECOME USER , CREATE ROLE , CREATE RULE , CREATE TYPE , CREATE USER , CREATE VIEW , ALTER SYSTEM , AUDIT SYSTEM , CREATE TABLE , DROP PROFILE ,ALTER PROFILE , ALTER SESSION , DROP ANY ROLE,

DROP ANY RULE,DROP ANY TYPE,DROP ANY VIEW ,QUERY REWRITE ,ALTER ANY ROLE , ALTER ANY RULE , ALTER ANY TYPE , ALTER DATABASE ,
CREATE ANY JOB , CREATE CLUSTER , CREATE LIBRARY , CREATE PROFILE , CREATE SESSION , CREATE SYNONYM , CREATE TRIGGER , DROP ANY INDEX , DROP ANY TABLE , GRANT ANY ROLE , LOCK ANY TABLE ,UNDER ANY TYPE ,UNDER ANY VIEW ,ALTER ANY INDEX , ALTER ANY TABLE , CREATE ANY RULE , CREATE ANY TYPE ,CREATE ANY VIEW ,CREATE OPERATOR ,CREATE RULE SET ,CREATE SEQUENCE ,DROP TABLESPACE , UNDER ANY TABLE ,ALTER TABLESPACE ,BACKUP ANY TABLE ,CREATE ANY INDEX , CREATE ANY TABLE , CREATE DIMENSION CREATE INDEXTYPE ,CREATE PROCEDURE ,DELETE ANY TABLE ,DROP ANY CLUSTER ,DROP ANY CONTEXT ,DROP ANY LIBRARY ,DROP ANY OUTLINE ,DROP ANY SYNONYM ,DROP ANY TRIGGER ,EXECUTE ANY RULE ,EXECUTE ANY TYPE ,INSERT ANY TABLE ,MANAGE ANY QUEUE ,MANAGE SCHEDULER ,SELECT ANY TABLE ,UPDATE ANY TABLE ,ALTER ANY CLUSTER ,ALTER ANY LIBRARY ,ALTER ANY OUTLINE ,ALTER ANY TRIGGER ,COMMENT ANY TABLE ,CREATE TABLESPACE
DEQUEUE ANY QUEUE ,DROP ANY OPERATOR ,DROP
ANY RULE SET ,DROP ANY SEQUENCE ,ENQUEUE ANY QUEUE ,EXECUTE ANY CLASS ,FORCE TRANSACTION ,MANAGE TABLESPACE ,ON COMMIT REFRESH ,ALTER ANY RULE SET ,ALTER ANY SEQUENCE,CREATE ANY CLUSTER ,CREATE ANY CONTEXT ,CREATE ANY LIBRARY , CREATE ANY OUTLINE ,CREATE ANY SYNONYM ,CREATE ANY TRIGGER ,DROP ANY DIMENSION ,DROP ANY DIRECTORY ,DROP ANY INDEXTYPE ,DROP ANY PROCEDURE

RESTRICTED SESSION

إذا أردت أن تعلم ما هي roles الموجودة و الصلاحيات على مستوى النظام التي تحويها :

```
select role , privilege from role_sys_privs;
```

إذا أردت أن تعلم ما هي roles الموجودة و الصلاحيات على مستوى الغرض التي تحويها :

```
select role , privilege from role_obj_privs;
```

ماذا لو أراد مستخدمنا jazeera أن يعرف ماهي الصلاحيات التي يملكها ؟

على مستوى النظام :

```
select * from user_sys_privs;
```

على مستوى الأغراض :

```
select *  from user_obj_privs;
```

إدارة الموارد Resources

الان سيكون لدينا جولة مع إدارة الموارد .. Resourcesو لكن لماذا ؟

تخيلوا السيناريو التالي :

لنفترض أنك مدير للقاعدة في إحدى البنوك أو شركات الاتصالات و تصادف وجود جدول بملايين السجلات) جدول الاتصالات الهاتفية التي يتم إجراءها) و جاء موظف جديد ليس له خبرة و نفذ تعليمة select أو Update بدون شرط where أو كانت نتيجة التعليمة التي نفذها تحوي على مليون أو أكثر من السجلات ؟ فما رأيك ؟ ما الذي سيحدث على السيرفر ؟ و هل ستتحمل الشبكة أو السيرفر هذا الضغط ؟ ماذا لو لم يكن موظفا واحداً و كان هناك مجموعة أخرى من الموظفين لن أقول أغبياء و لكن مستهترين و لا يبالون ماذا سوف يحدث ؟ لأنك الوحيد الذي سيلام على توقف القاعدة...

ما الحل ؟؟

Resource Limit Profiles :

هي مجموعة من القيم التي نضعها لكي نتحكم بحجم الذاكرة أو كمية البيانات التي يستطيع المستخدم استغلالها ضمن جلسته الحالية ، حيث يتم وضع هذه القيم داخل ما نسميه Profile و يتم إسناده إلى مستخدم أو أكثر عن طريق تعليمات create userأو . alter user

ليتم تفعيل هذه الإدارة نحتاج لأن نضع القيمة TRUE إلى البارامتر resource_limit بالطريقة التالية :

```
Enter user-name: system/oracle
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> alter system set resource_limit=true;
System altered.
```

و الآن لنذكر البارامترات التي نستطيع استخدامها :

1- cpu_per_sesison و فيها نحدد الوقت الزمني المخصص لكل جلسة و التي تستطيع استهلاكه من زمن المعالج ، يقاس هذا الوقت بالثانية .

2- session_per_user كم عدد الجلسات التي يمكن لمستخدم ما أن يقوم بفتحها . (يعلم الجميع بأننا نستطيع تشغيل SQL plus أكثر من مرة و بنفس اسم المستخدم ، كل نسخة من sql plus مفتوحة ندعوها جلسة ..)

3- connect_time كم الوقت الزمني الذي تسمح به للمستخدم للإتصال بالقاعدة 4- IDLE_TIME الوقت الزمني الذي ستمح به للمستخدم أن يبقي جلسته مفتوحة دون أن يقوم بأي شيء .

5- LOGGICAL_READS_PER_SESSION كم كمية البيانات التي تريد لمستخدم ما أن يقوم بقراءتها خلال الجلسة الكاملة .. تقاس هذه الكمية بعدد بلوكات البيانات .. Data Block و لكي نفهمها بطريقة مبسطة لإغنها المعادلة التالية : العدد * ٤ كيلو بايت (في العادة ... (

6- Private_SGA و هي المساحة المخصصة من الذاكرة لكل جلسة يقوم بفتحها.

7- CPU_PER_CALL و هو الزمن الأقصى الذي يسمح لتعليمة المستخدم أن تأخذ من زمن المعالج .

LOGICAL_READS_PER_CALL : كم كمية البيانات التي تريد لمستخدم ما أن يقوم بقراءتها خلال التعليمة الواحدة .

لنتقل إلى التطبيق العملي :

تعليمة إنشاء : Profile

```
CREATE PROFILE profile LIMIT
[SESSIONS_PER_USER max_value]
[CPU_PER_SESSION max_value]
[CPU_PER_CALL max_value]
[CONNECT_TIME max_value]
[IDLE_TIME max_value]
[LOGICAL_READS_PER_SESSION max_value]
[LOGICAL_READS_PER_CALL max_value]
[COMPOSITE_LIMIT max_value]
[PRIVATE_SGA max_bytes]
where:
profile is the name of the profile
max_value is an integer, UNLIMITED, or DEFAULT
max_bytes is an integer optionally followed by K or M
UNLIMITED, or
DEFAULT
```

ما رأيكم بالمثال التالي :

سنسمح للمستخدم jazeera بأن يفتح جلستين فقط و سنسمح له بأن يستهلك من وقت المعالج في التعليمة الواحدة

دقيقتين فقط و سنسمح له بقراءة ١ ميغا بايت فقط من البيانات في التعليمة الواحدة و سنسمح له بفتح جلسة فقط لمدة ثلاثين دقيقة :

```
SQL> create profile jazeera _prof limit
            sessions_per_user     2
            cpu_per_call       120
            logical_reads_per_call    1073741824
            connect_time 1800;

Profile created.
SQL> alter user jazeera profile jazeera _prof;

User altered.
```

إدارة كلمة السر:

تعتبر إدارة كلمات السر في أوراكل مشابهة لإدارة كلمات السر في في الويندوز ، فهي تتحكم بعمر كلمة السر (أي الفترة الزمنية التي يجب على المستخدم تغيير الكلمة بعدها) أو هل يجب تكرار الكلمة أو عدد الأحرف الصغرى لها ..... إلخ و تهدف هذه الإدارة إلى جعل المستخدمين يلتزمون بقواعد تكفل لكلمات مرورهم السلامة و تقليل احتمالات سرقتها من قبل أعضاء آخرين ... إلخ .

يتم التعامل مع إدارة كلمات السر كما تعاملنا سابقاً مع إدارة الموارد فلا بد من وجود **profile** نضع فيه الإعدادات التي نريد للتحكم ..

ما هي **parameter** التي نستطيع استخدامها :

1- **FAILED_LOGIN_ATTEMPTS** عدد مرات التي سيخطئ فيها المستخدم بإدخال كلمة السر و عندها سيتم إغلاق الحساب .

2- **PASSWORD_LOCK_TIME** عدد أيام الفترة الزمنية للإغلاق ..

3- **PASSWORD_LIFE_TIME** عدد الأيام التي يجب على المستخدم أن يقوم بتغيير كلمة مروره بعدها ..

4- **PASSWORD_GRACE_TIME** ما هي عدد الأيام التي تريد إعطاءها للمستخدم حتى يقوم بتغيير كلمة سره بعد انقضاء الفترة التي حددناها في المثال السابق ..

5- **PASSWORD_REUSE_TIME** عدد الأيام التي ستسمح للمستخدم بعدها أن يعيد استخدام كلمة مرور قديمة..

٦- **PASSWORD_REUSE_MAX** : عدد المرات التي يمكن استخدام كلمة السر القديمة ، ويمكن اجبار المستخدم بعدم استخدام كلمة سر قديمة ابدأ بوضع قيمة المعامل صفر.

ملاحظة :

لا يمكن تحديد قيمة للعاملين (٥,٦) معاً إذا تم تحديد قيمة أحد العاملين يجب و ضع قيمة الاخر **UNLIMITED**.

7- **PASSWORD_VERIFY_FUNCTION**: لتحديد ال **PL/SQL function** التي سوف تستخدم في وصع قوانين على اختيار كلمة السر.

ملاحظة:

- يمكن و ضع كلمة قوانين لتسمية كلمة السر باستخدام **PL/SQL** من هذه القوانين التي يمكن تحديدها أن لا تكون كلمة السر مطابقة لاسم المستخدم أو يجب أن تحتوي كلمة السر على أرقام وأحرف ...الخ
- يجب تكوين الدالة في **SYS Schema** وتكون من نوع **Boolean**.

للتوضيح : لنفرض أن المستخدم **jazeera** وضعنا له القيم التالية :

FAILED_LOGIN_ATTEMPS 3

PASSWORD_LOCK_TIME 2

هذا يعني : عندما يحاول تسجيل الدخول بكلمة سر خاطئة أكثر من ثلاثة مرات متتالية عندها سيتم قفل حسابه لمدة يومين إثنين (إلا إذا قام المدير بفك قفل الحساب يدوياً) قبل أن يتمكن من تسجيل الدخول مرة أخرى ..

إذا وضعنا القيم التالية :

PASSWORD_LIFE_TIME 15

PASSWORD_GRACE_TIME 2

PASSWORD_REUSE_TIME 15

لنفرض أننا الآن يوم ٢٠٠٦/٧/٣١ و كلمة المرور الآن pass1 عندها :

في اليوم ٨/١٤ يجب على المستخدم أن يغير كلمة السر بحيث لديه مهلة حتى ٨/١٦ لعملية التغيير و لن يستطيع استعمال كلمة سره القديمة pass1 بل يجب أن يضع كلمة مرور جديدة و لن يستطيع إعادة كلمة السر القديمة (pass1 أي يعيد استخدامها) إلا بعد التاريخ ٨/٣١ ... أرجو أن تكون الفكرة قد وصلت.

CREATING PROFILE TO MANAGE PASSWORD:

يمكن تكوين profile لإدارة كلمة السر بتباع القاعدة التالية:

```
CREATE PROFILE profile_name LIMIT
[FAILED_LOGING_ATTEMPTS number]
[PASSWORD_LOCK_TIME number_of_days]
[PASSWORD_LIFE_TIME number_of_days]
[PASSWORD_GRACE_TIME number_of_days]
[{PASSWORD_REUSE_TIME | PASSWORD_REUSE_MAX}
number_of_days]
 [PASSWORD_VERIFY_FUNCTION {function_name | NULL | DEFAULT}]
```

مثال :

```
CREATE PROFILE myprofile LIMIT
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_LOCK_TIME 5
PASSWORD_LIFE_TIME 15
PASSWORD_GRACE_TIME 7
PASSWORD_REUSE_TIME 15;
```

ملاحظة :لاستخدام الساعات أو الدقائق عوضاً عن الأيام تستخدم القيمة 1/24 للدلالة على ساعة واحدة ويستخدم 1/1400 للدلالة على دقيقة واحدة .

ALTERING PROFILE:

```
ALTER PROFILE myprofile limit
FAILED_LOGIN_ATTEMPTS 2
PASSWORD_LOCK_TIME UNLIMITED
PASSWORD_LIFE_TIME 10
PASSWORD_GRACE_TIME 7/24
PASSWORD_REUSE_TIME 10;
```

ما رأيكم بمثال :

دعونا نجبر مستخدمنا jazeera على أن تكون كلمة سره تبدأ ب 'j'و أن لا تكون مثل إسمه أي jazeeraأو أن لا تساوي 'pass' :

```
SQL> connect sys/oracle as sysdba
     Connected
```

```
create or replace function Check_pass(userid_param in varchar2 , pass in varchar2 , old_pass
varchar2) return boolean
 as
   ch char;
   begin
 ch:=substr(pass,1,1);
 if  ( ch='j') and (pass <> 'pass' ) and  (pass<>' jazeera'  )  then
return true;
   else
return false;
  end if;
   end;
/
Function created.
SQL> create  profile jazeera  _pro
  2  limit password_verify_function Check_pass;
SQL> alter user jazeera
  2  profile jazeera  _prof;

SQL> alter user jazeera
  2  password expire;

SQL> connect jazeera  / jazeera  ;
ERROR:
ORA-28001: the password has expired
Changing password for jazeera
New password: jazeera
Retype new password: jazeera
ERROR:
ORA-28003: password verification for the specified password failed
ORA-28003: password verification for the specified password failed
Password unchanged
```

:DROPPING PROFILE

```
DROP PROFILE profile_name [CASCADE];
```

مثال : حذف profile الذين خصص لهم و تحويلهم الى Default

```
DROP PROFILE myprofile CASCADE;
```

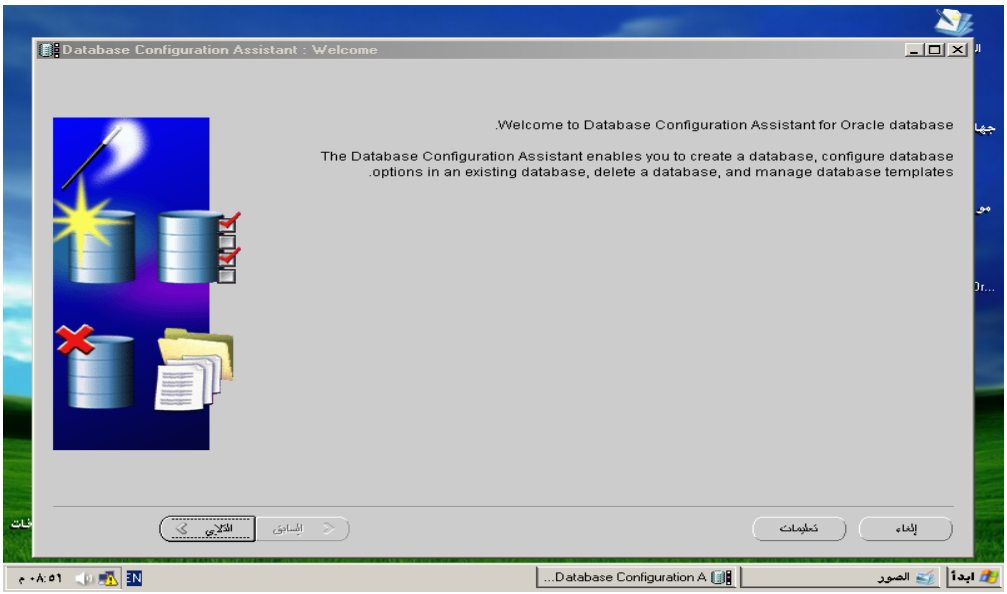
إنشاء قاعدة جديدة :

سنستخدم في إنشاء القاعدة أداة رائعة و هي : (Database Configuration Assistants) : DBCA
تضمن هذه الأداة إنشاء قاعدة جديدة أو إزالة قاعدة بيانات موجودة أو تعديل قاعدة بيانات ... إلخ .
لنفذ الخطوات التالية :

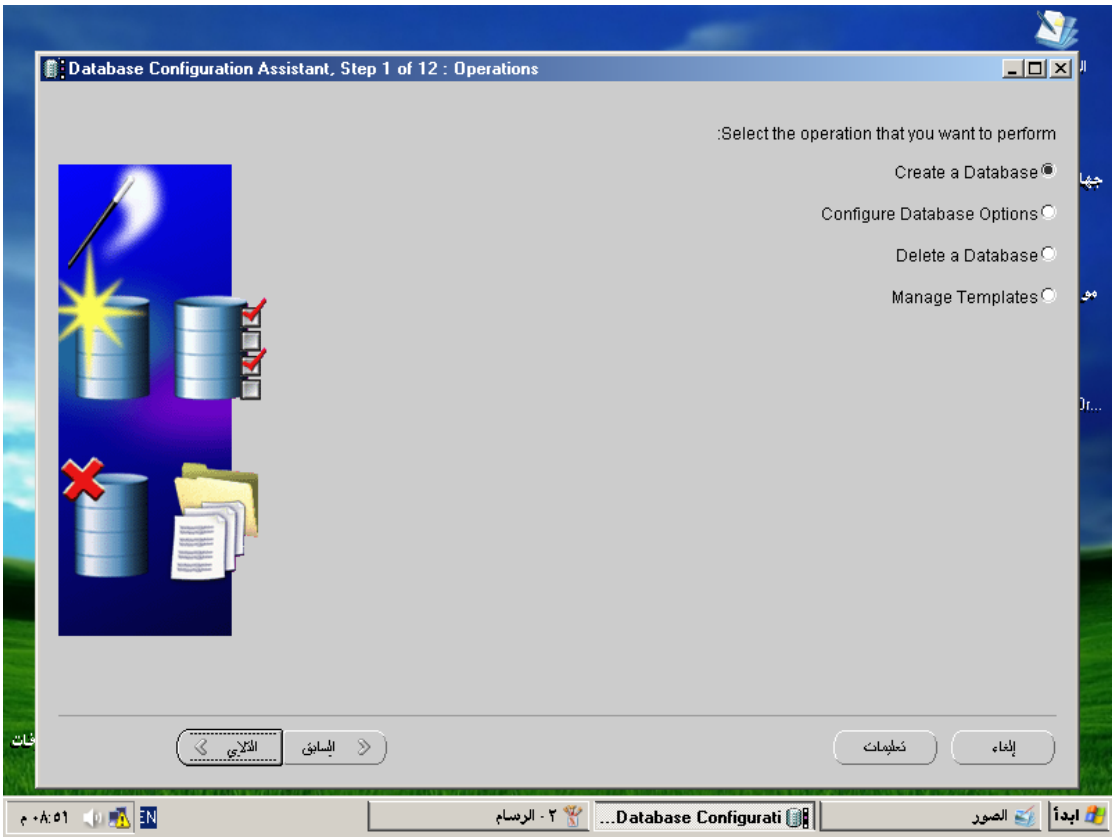
-1من

start->programs ->Oracle10g->Configuration and Migration Tools -> Database Configuration Assistants .

تظهر الواجهة مرحبة بنا على طريقة أوراكل (بدون القهوة العربية) نختار . Next



2-نختار الخيار الأول Create Database .



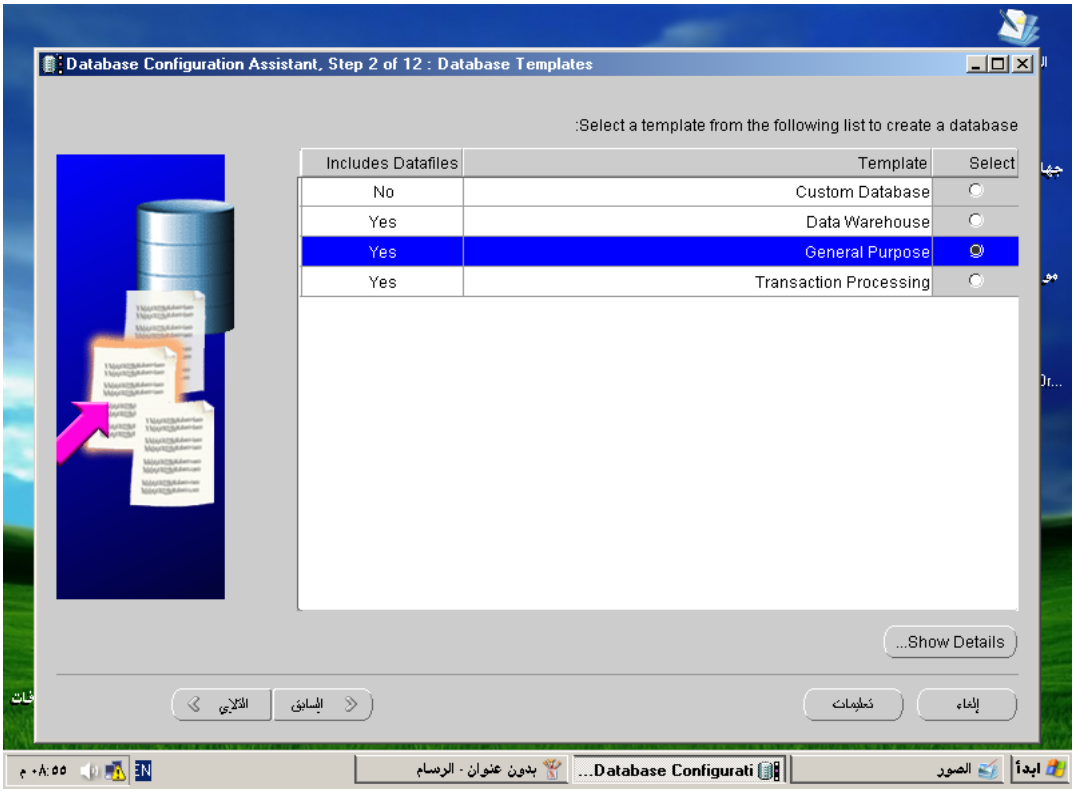
3- تظهر لنا أنواع من قواعد البيانات دعوني أشرحها:

١- Data Ware House : .هذه النوع موجه نحو القواعد التي تحوي على كميات ضخمة من البيانات التي تضمن سرعة خارقة في جلب البيانات من جداول قد تحوي بلايين من السجلات ... لهذا النوع أبحاث كثيرة نظراً لأهميته البالغة.

٢- Transaction Processing : و هي موجهة للقواعد التي تخضع لضغط عالي من التعديلات و الإدخالات الجديدة و تضم القليل من عمليات إحضار البيانات ..

٣- General Purpose : موجهة إلى القواعد العادية ذات الغرض العام (كما في حالتنا مثلاً)

لذلك نختار General Purpose ثم التالي..

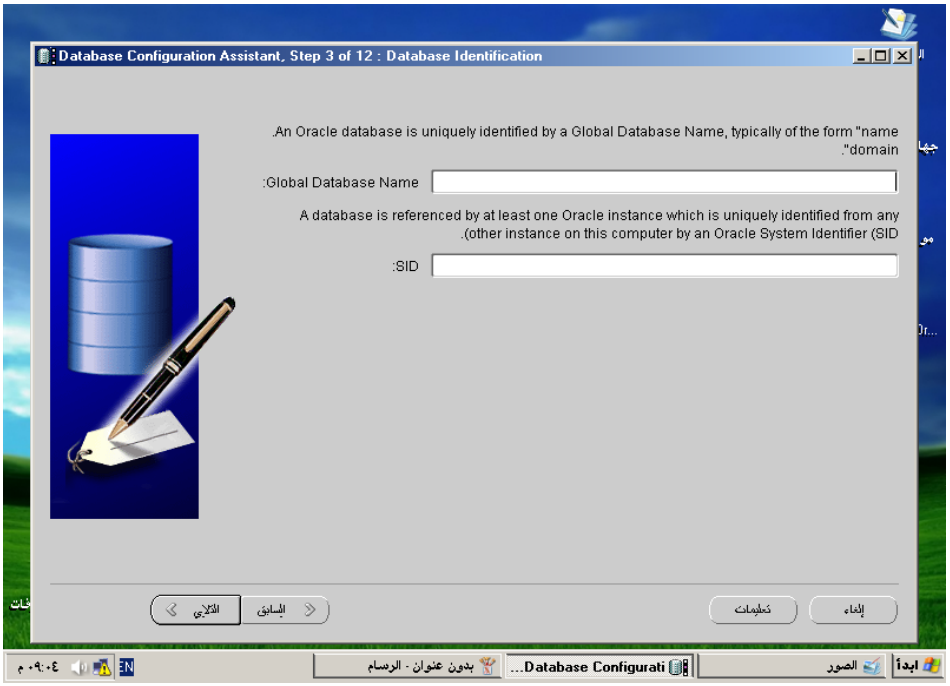


4- لدينا قيمتين :

١- Global Database Name اسم قاعدة البيانات :

٢- SID اسم instance الذي سيتعامل مع هذه القاعدة :

للتذكير فقط : في الوضع الافتراضي فإنك عندما تنصب أوراكل فإنه ينشئ قاعدة بيانات و instance اسميها متطابقين أي ORCL

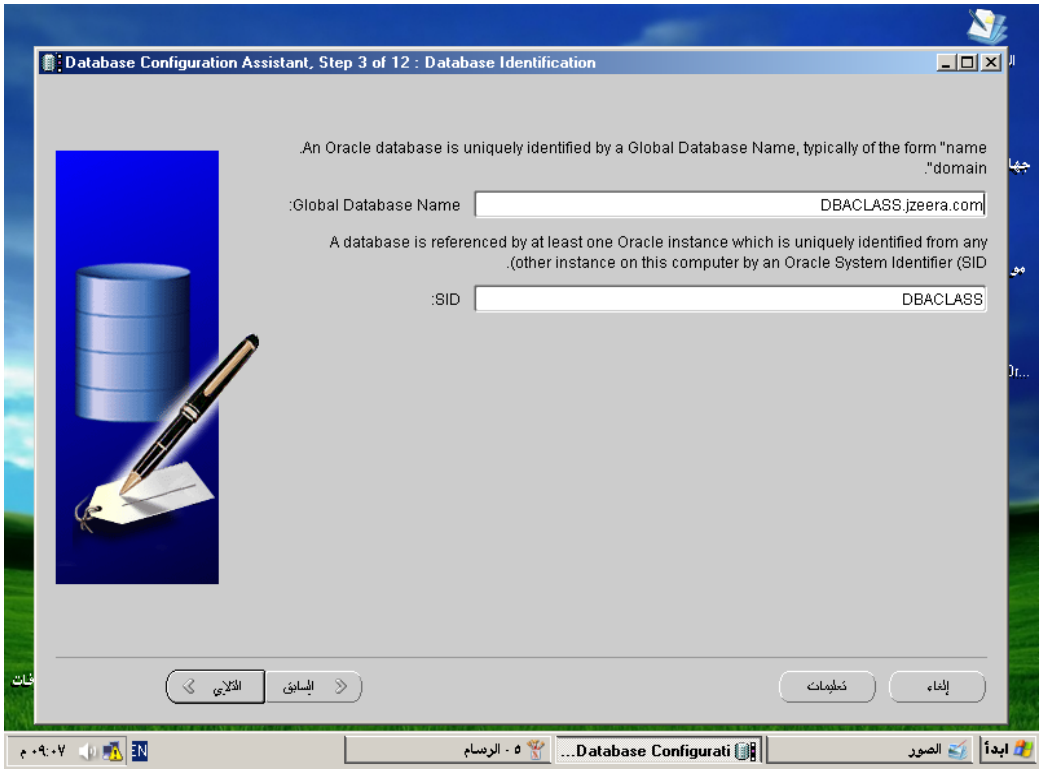


لنتفق على اسم واحد لاستعماله :

Global Database Name : DBACLASS.jzeera.com

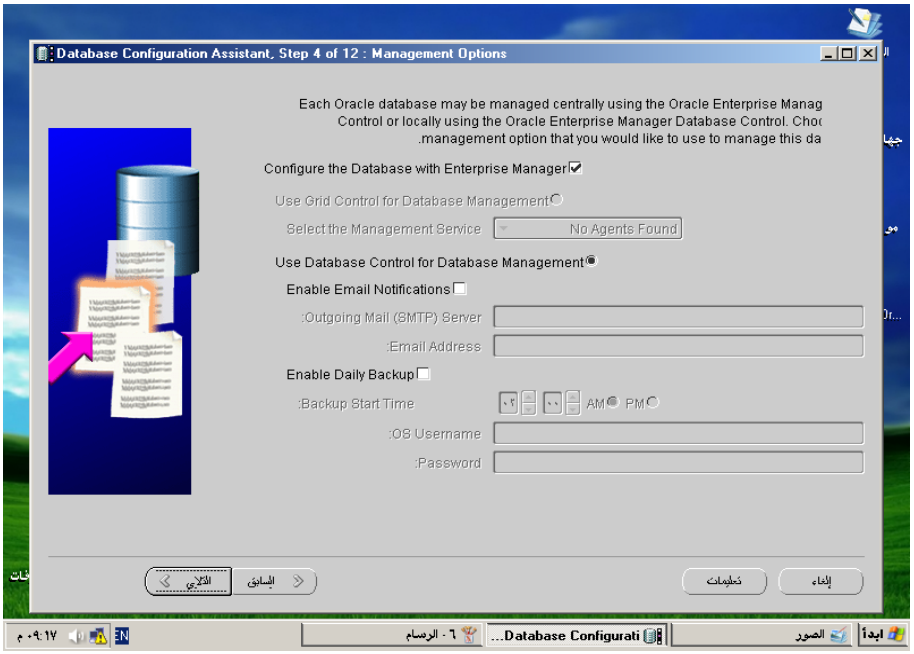
SID : DBACLASS

ثمNext

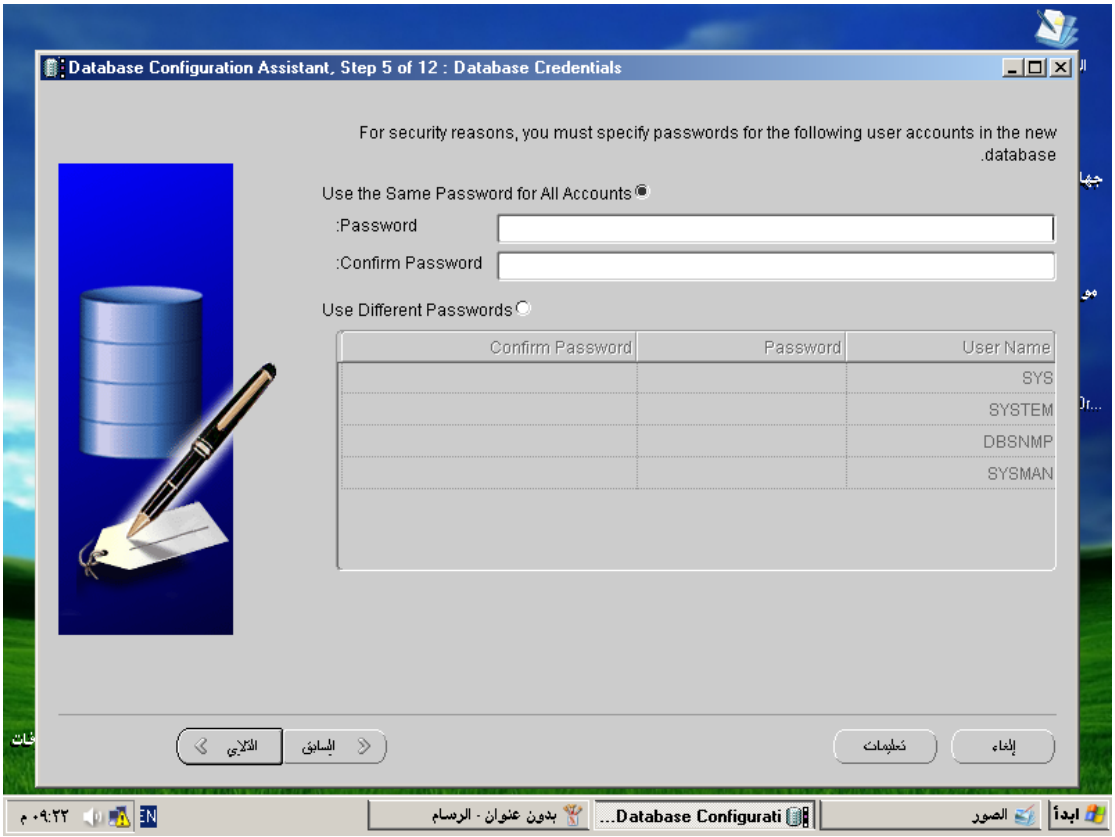


5- يوجد عدد من الخيارات :

لنتأكد من أن الخيار Configure the database using enterprise manger مفعلاً حين أن Enterprise Managerهي أداة رائعة تمكننا من إدارة القاعدة بواجهة تفاعلية ممتازة .. ثم .. Next

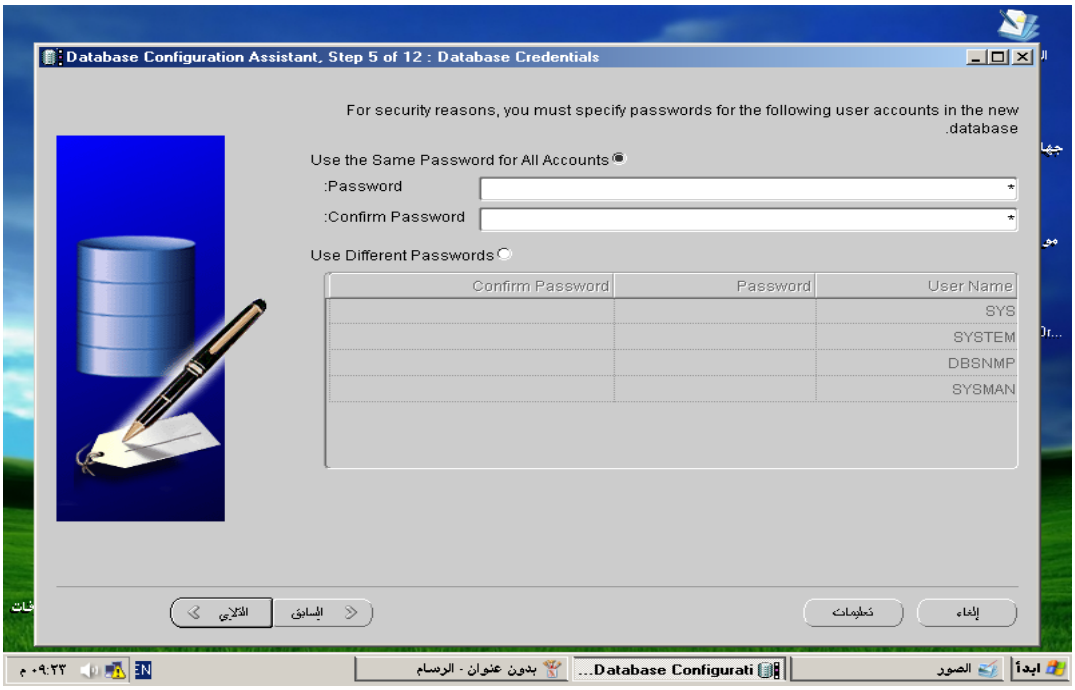


6- طلب منا وضع كلمة السر واحدة للمستخدمين SYSMAN , System , Sys: أو أن نضع لكل مستخدم كلمة مرور خاصة به...



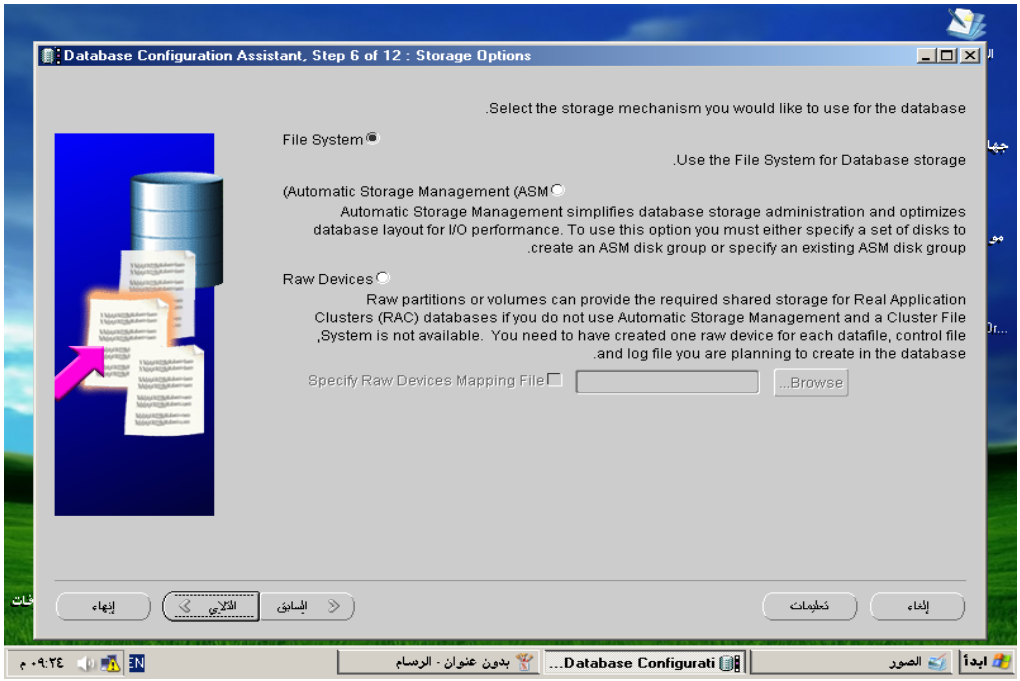
نضع كلمة سر موحدة و لتكن (b)

ثم .. Next



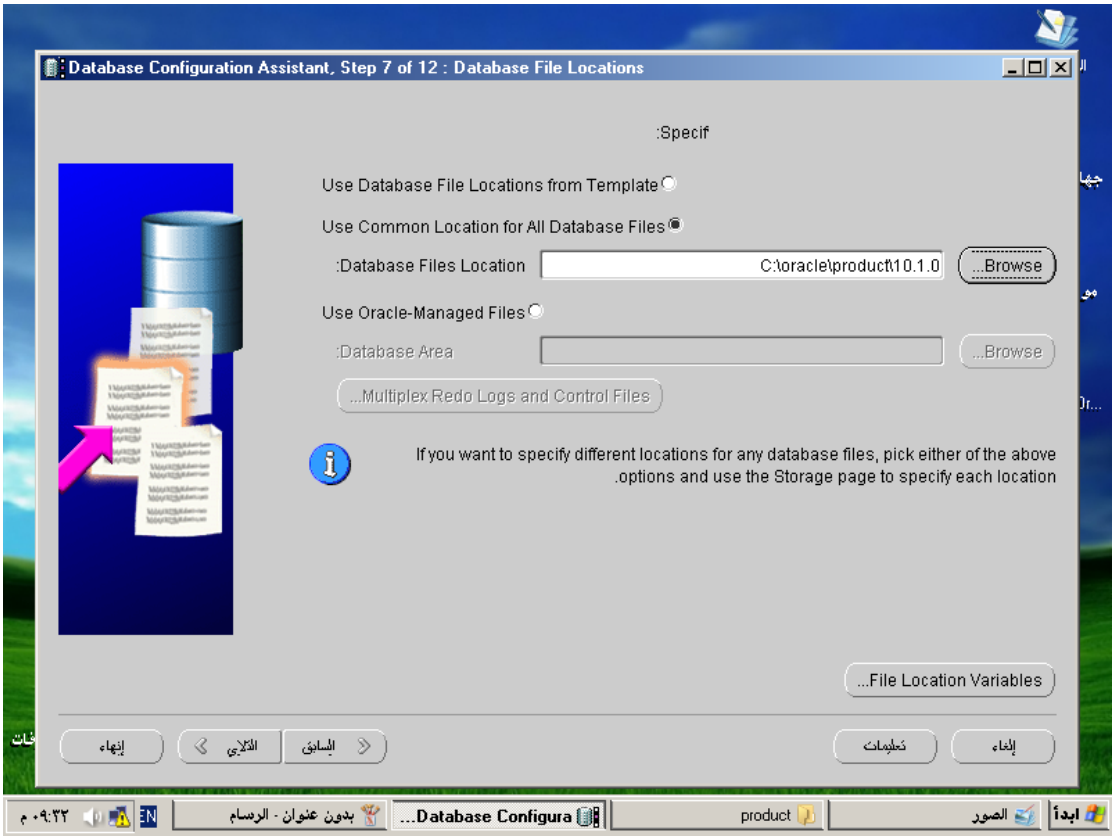
7- نختار File System : أي نظام الملفات الحالي الذي يستخدمه نظام التشغيل

ثم .. Next ...



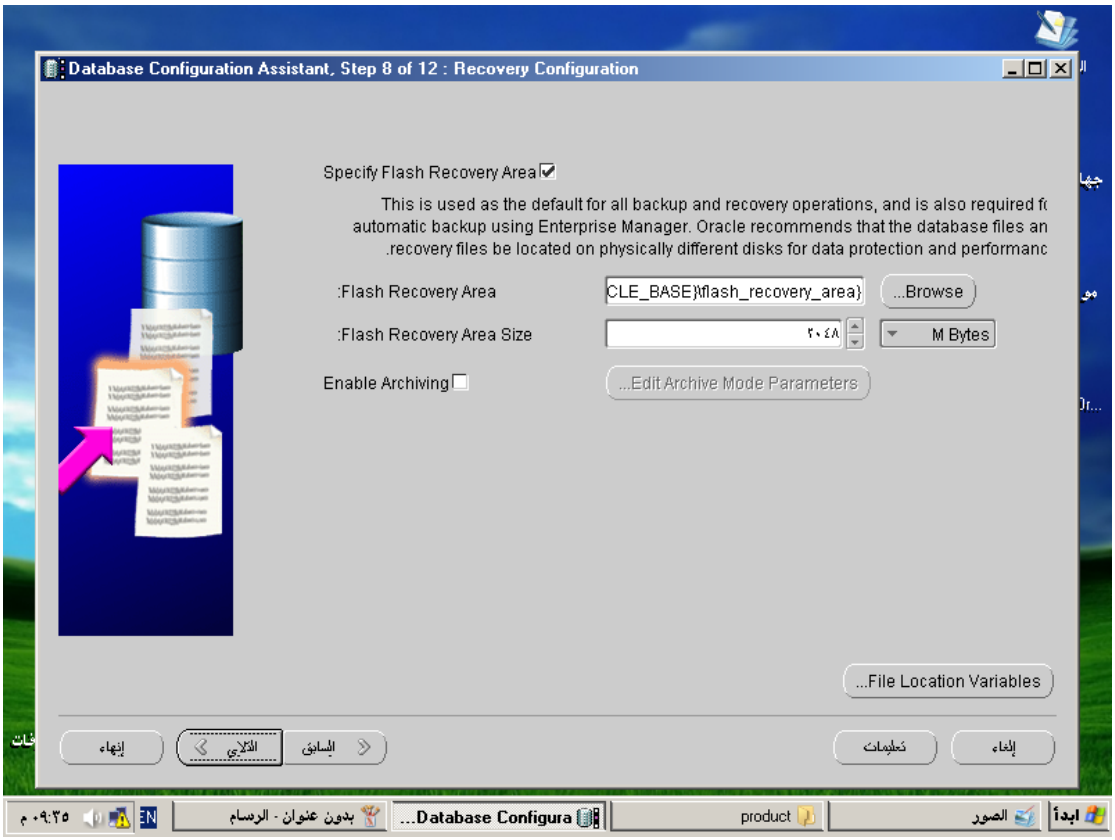
8- نختار توضع ملفات القاعدة:

١- use file allocations from Template: هنا يجب وجود إعدادات مسبقة قمنا بتجهيزها من قبل داخل Template ..
٢- Use Common Location for all Database Files : هنا نحدد مسار مجلد ما ليكون الحاوي لجميع ملفات القاعدة ..
٣- Use Oracle-Manged Files : هناك طريقة نسميها OMF و هي طريقة Standard تتبعها أوراكل في تسمية الملفات الخاصة بالقاعدة أي أن DBA يتنحى جانباً و تقوم أوراكل بتسمية الملفات و تحديد مواقعها بدلاً عنه ...
لنختر الخيار الثاني ثم .. Next



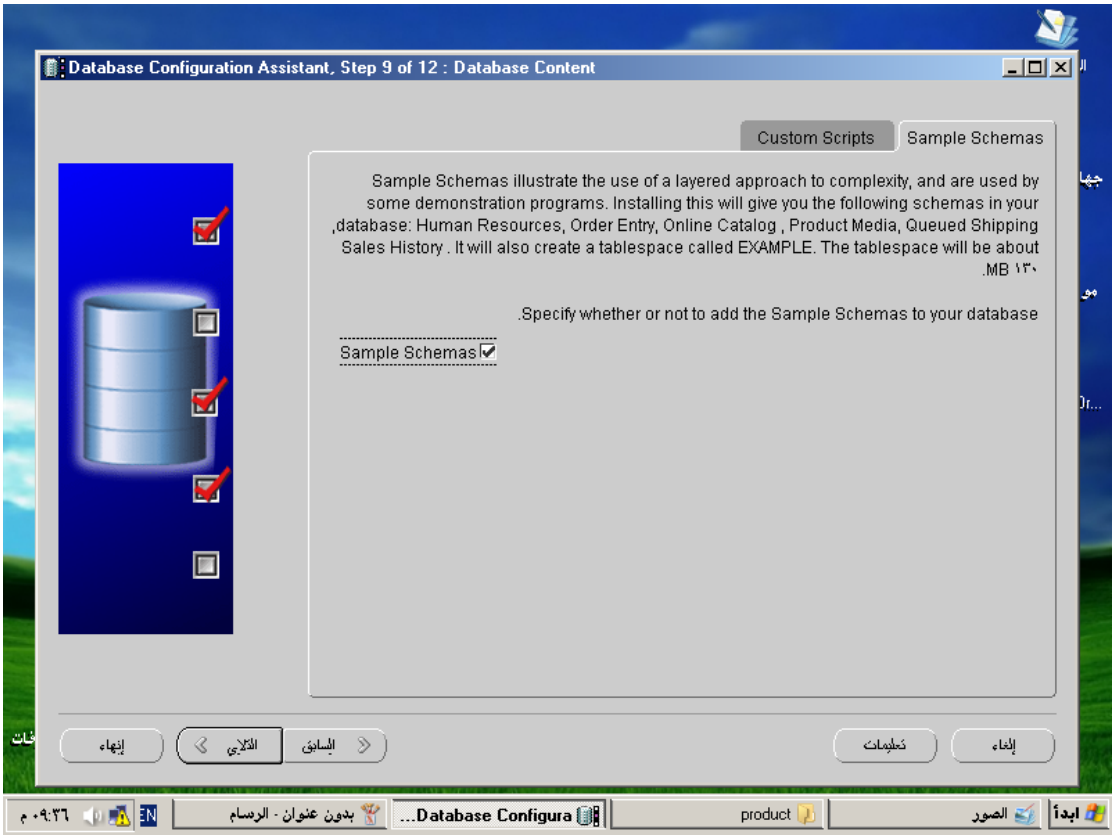
9- تظهر لنا خيارات النسخ الاحتياطي للقاعدة Backup سواء بالنسبة إلى Flash Recovery من أروع الإضافات التي شهدها الإصدار 10gأو .. Archiving Redo log

نضغط على .. Next



10- -10تظهر واجهة يمكننا اختيار Sample Schema لتثبيت مخططات مثل ... Scott-hr

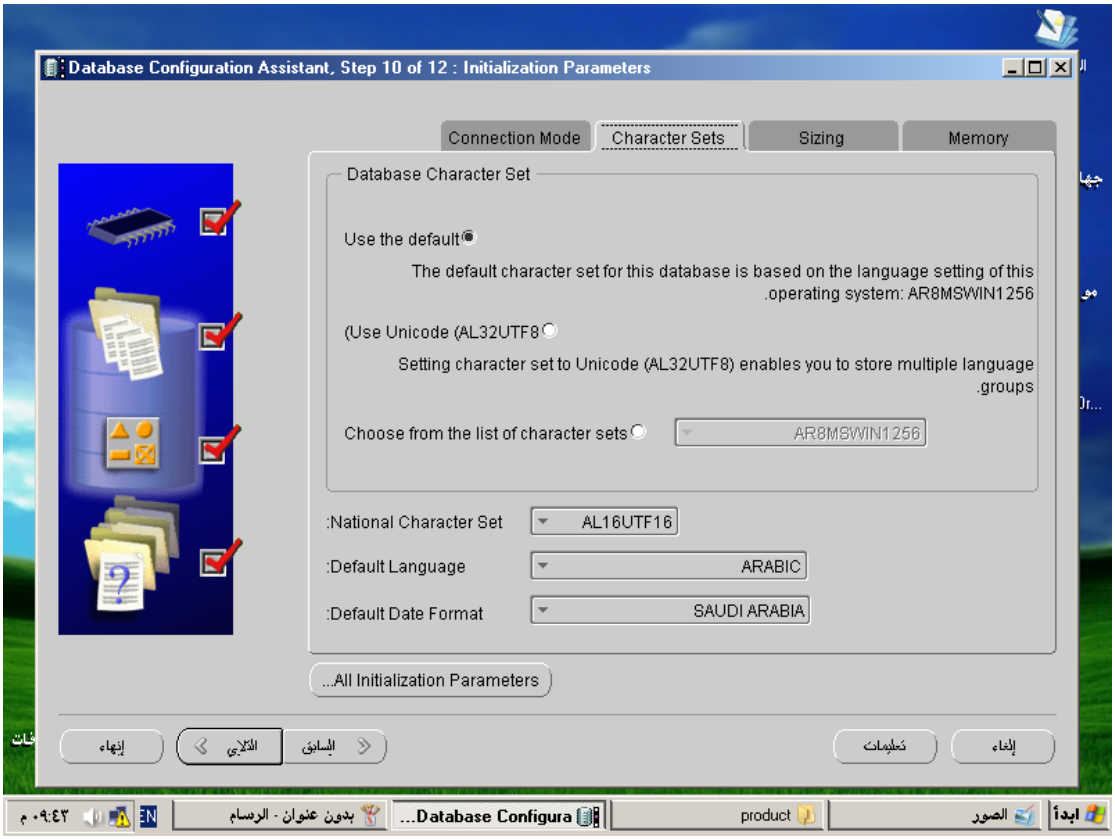
نختار.. Next



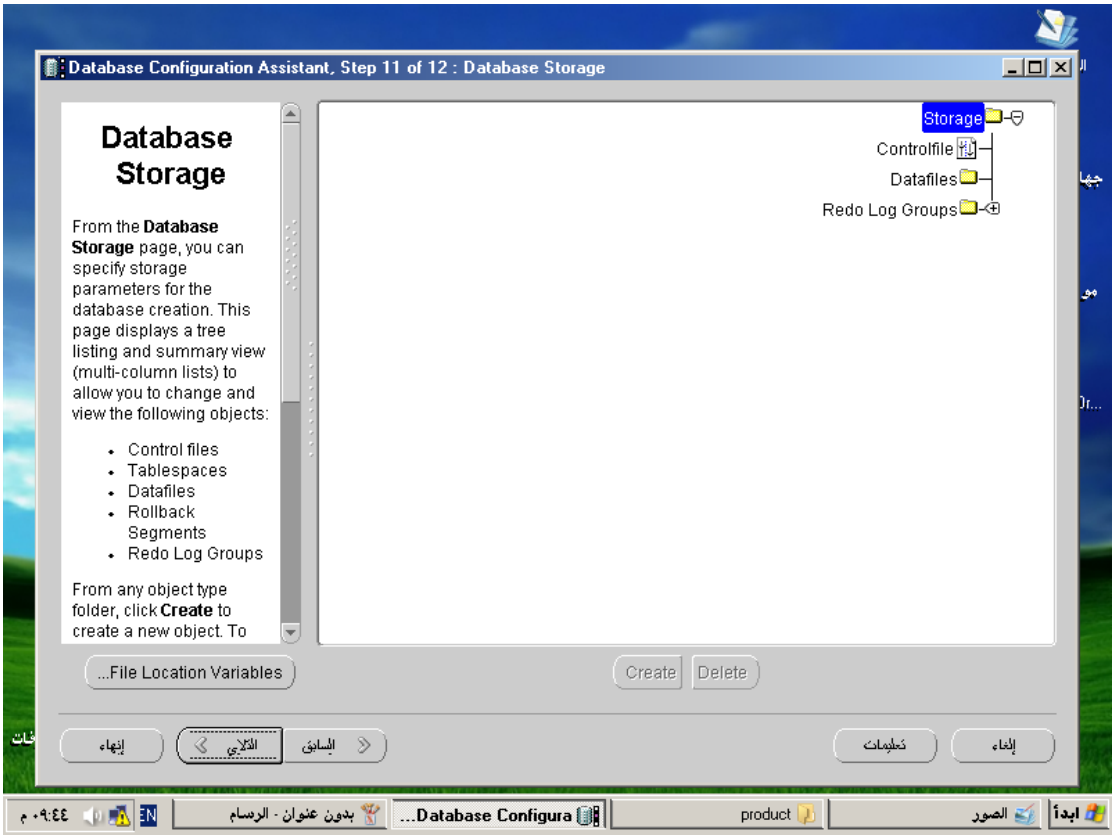
11- نحدد بارامترات خاصة ب... Oracle instance

ما يهمنا هنا هو character set و هي الترميز الذي سيتم استخدامه في تخزين البيانات داخل القاعدة و هو يلعب دوراً كبيراً في مسألة تخزين البيانات العربية أو لغات الشرق الأقصى مثل اليابانية و غيرها ... المهم بالنسبة لنا أن يكون هذا الترميز مناسباً للبيانات العربية لذلك من القسم

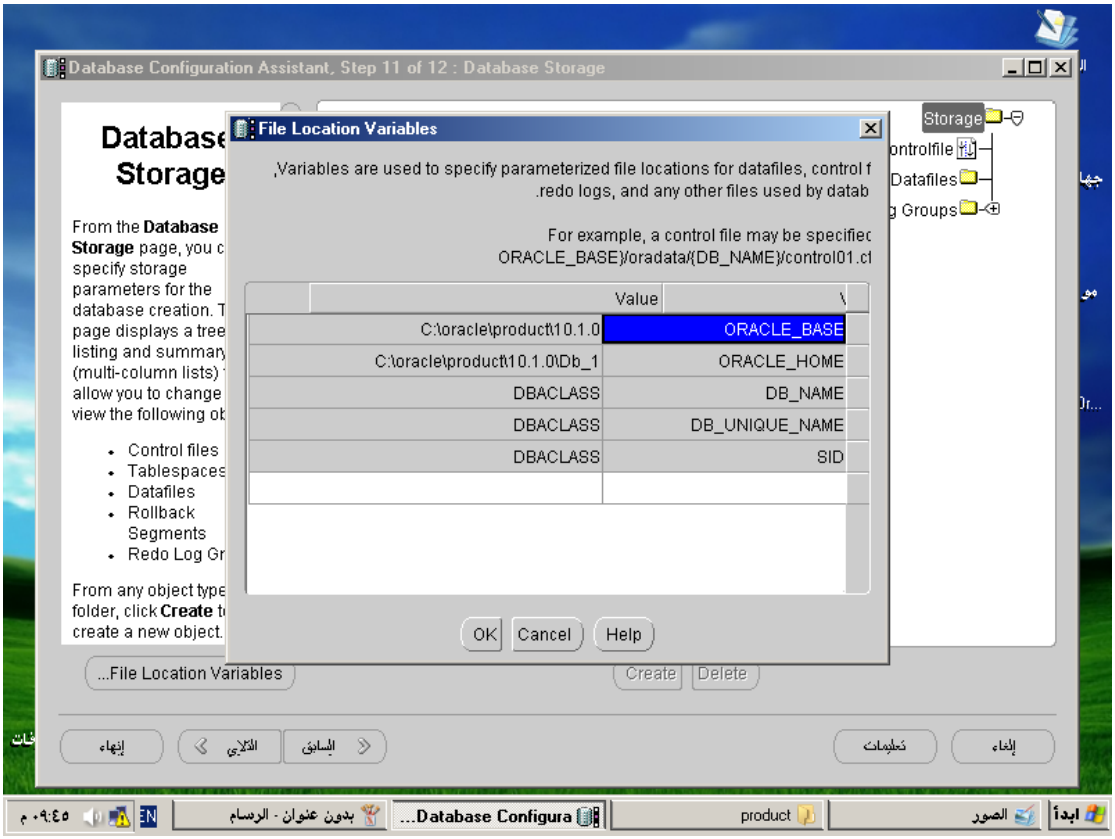
Database Character Set يمكننا :
1-اختيار Using Unicode: من أجل الترميز Unicode الذي يلائم لغتنا العربية ..
2-اختيار : Choose from a list of character setو بوسعنا إختيار هنا AR8MSWIN1256 من أجل العربية ...
من الخيارين Default Language و Default Data Format نحدد بهما لغة الرسائل و تنسيق التاريخ ... ثم .. Next



12- تظهر واجهة مهمة : إضغط على الزر File Allocation Variable : ستجد هناك مجموعة من المسارات المخزنة داخل متحولات .. SID .. Oracle Home .. Oracle_Base : إلخ يجب أن تحفظ هذه المسارات عن ظهر قلب لأنها مواقع الملفات التي سنتعامل معها لاحقاً ...



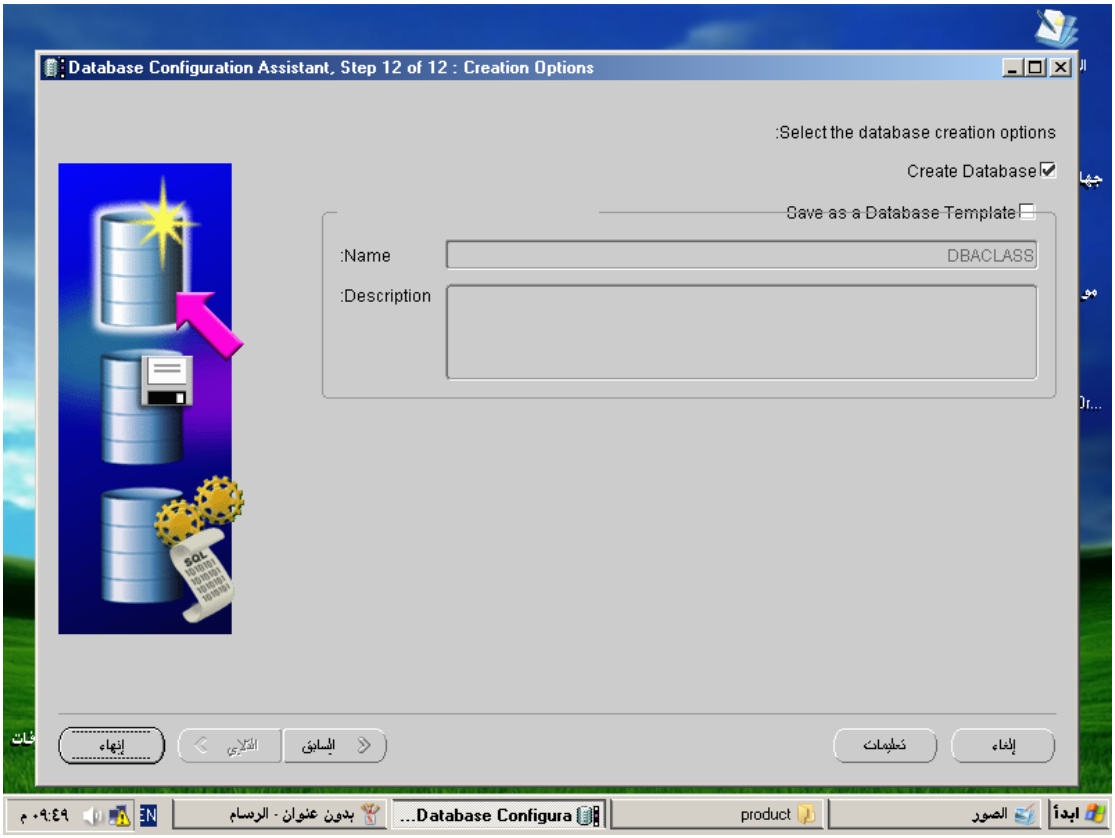
اخرج من هذه الواجهة ثم .. next



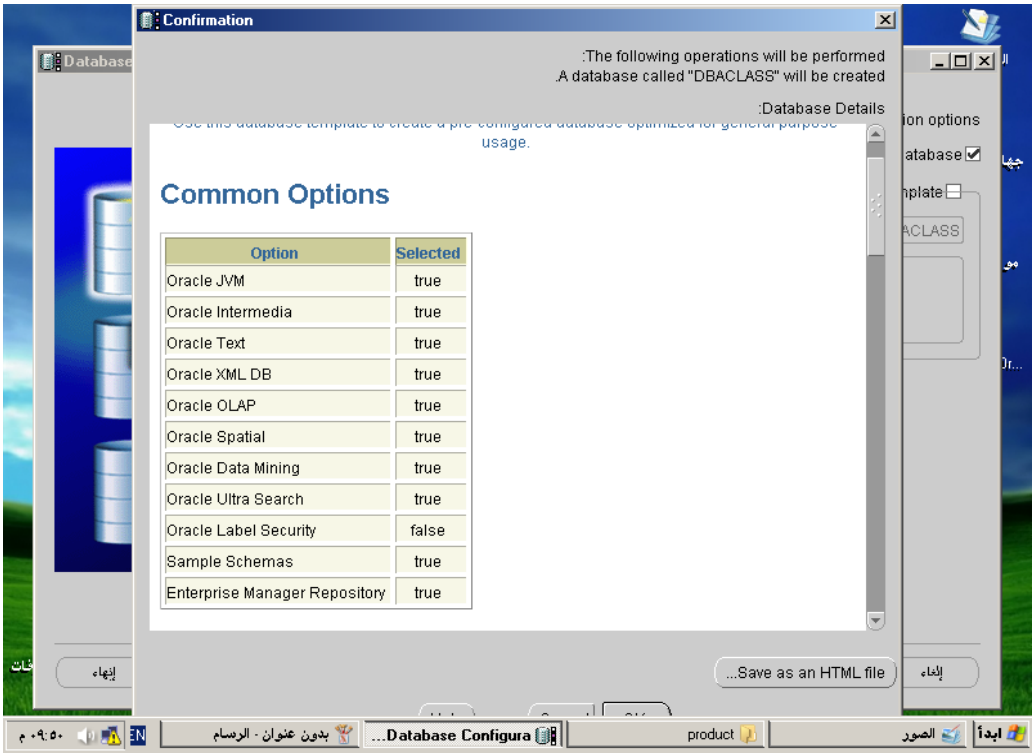
13-تظهر لك خيارات:

١- create Database : من أجل إنشاء قاعدة جديدة و هذا ما نريده

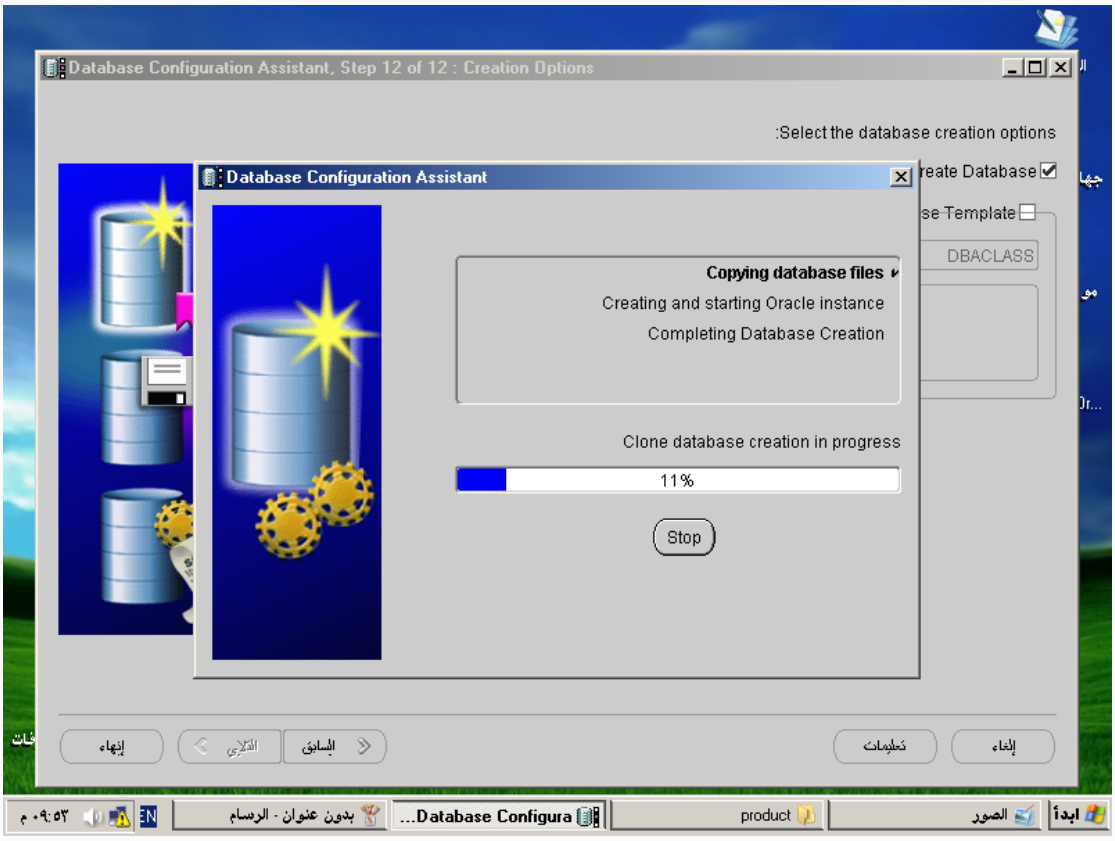
٢- save As Database Template : أي نحفظ الإعدادات التي نقد نستعملها لاحقاً في إنشاء قاعدة أخرى

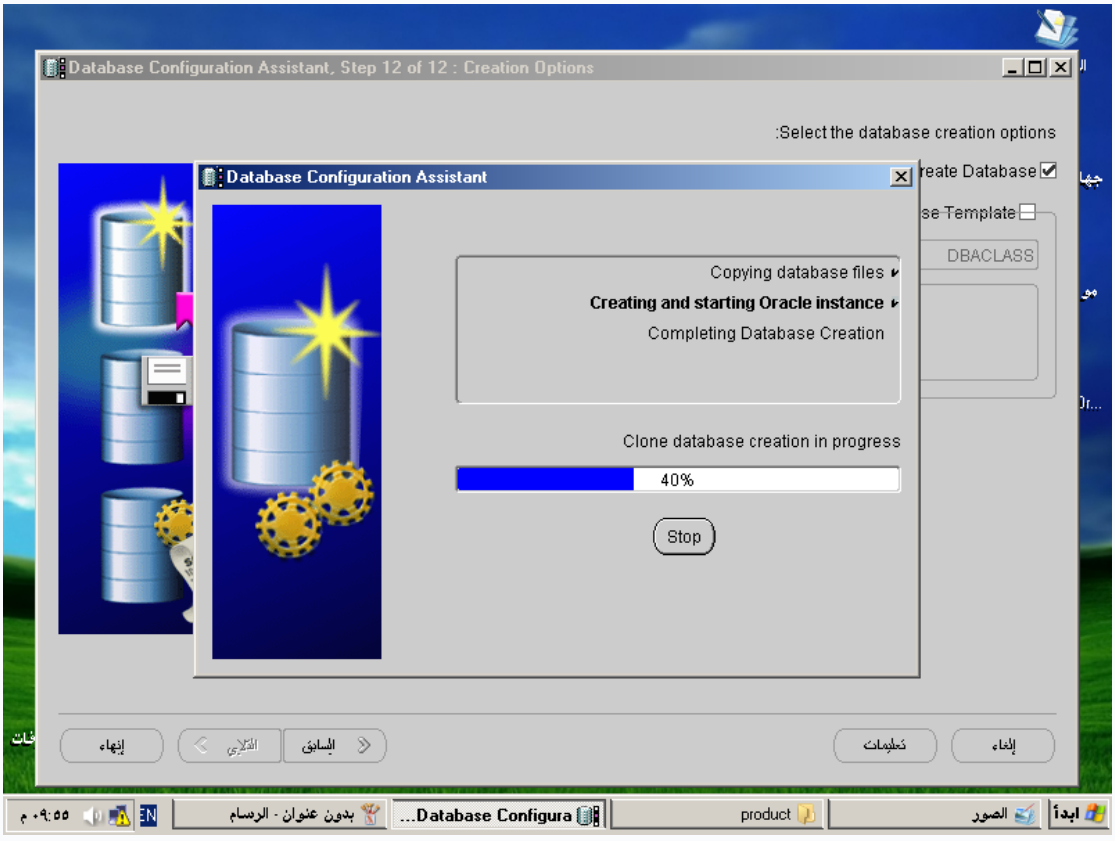


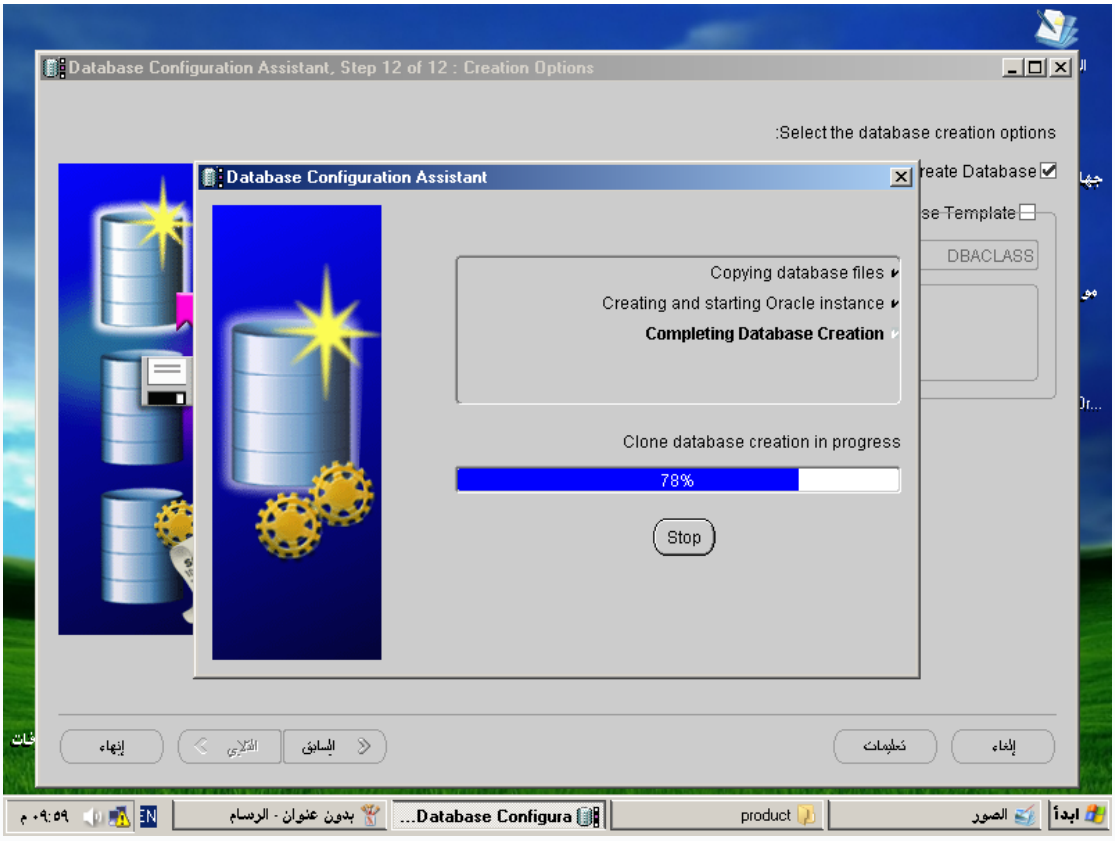
نضغط على زر finish



اضغط على زر OK للبدء في إنشاء القاعدة

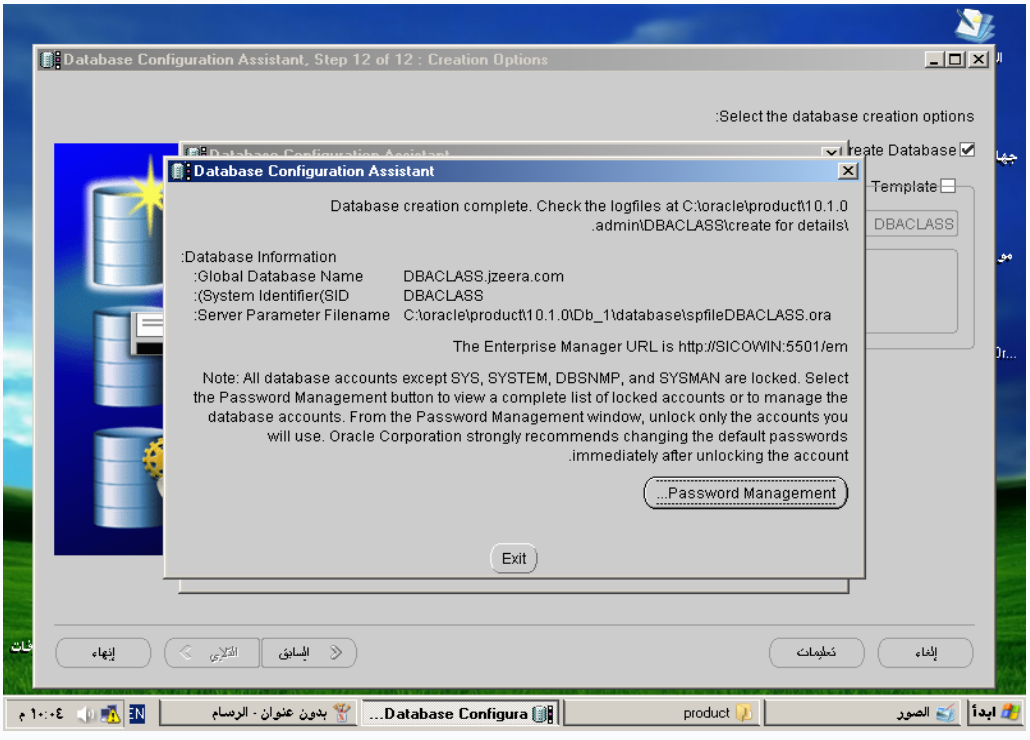






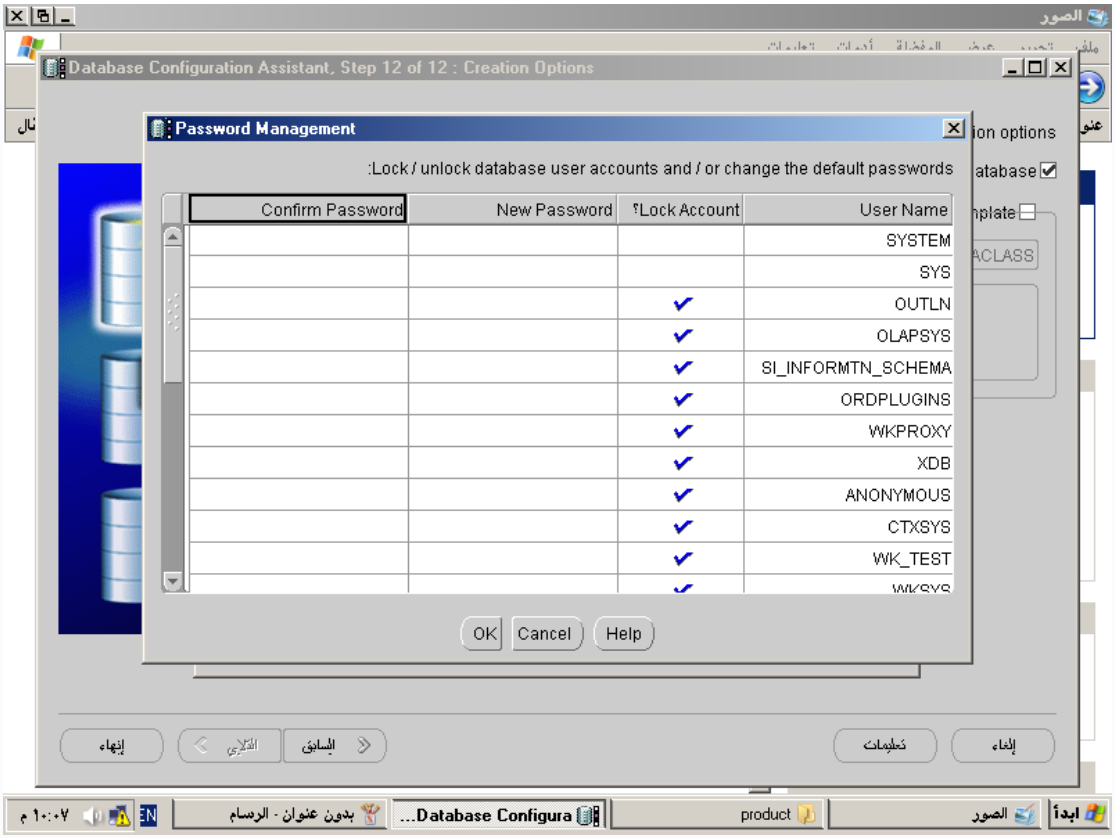
تظهر و اجهه للتعديل كلمات السر و فتح أو أغلاق الحساب

اضغط **Password Management**

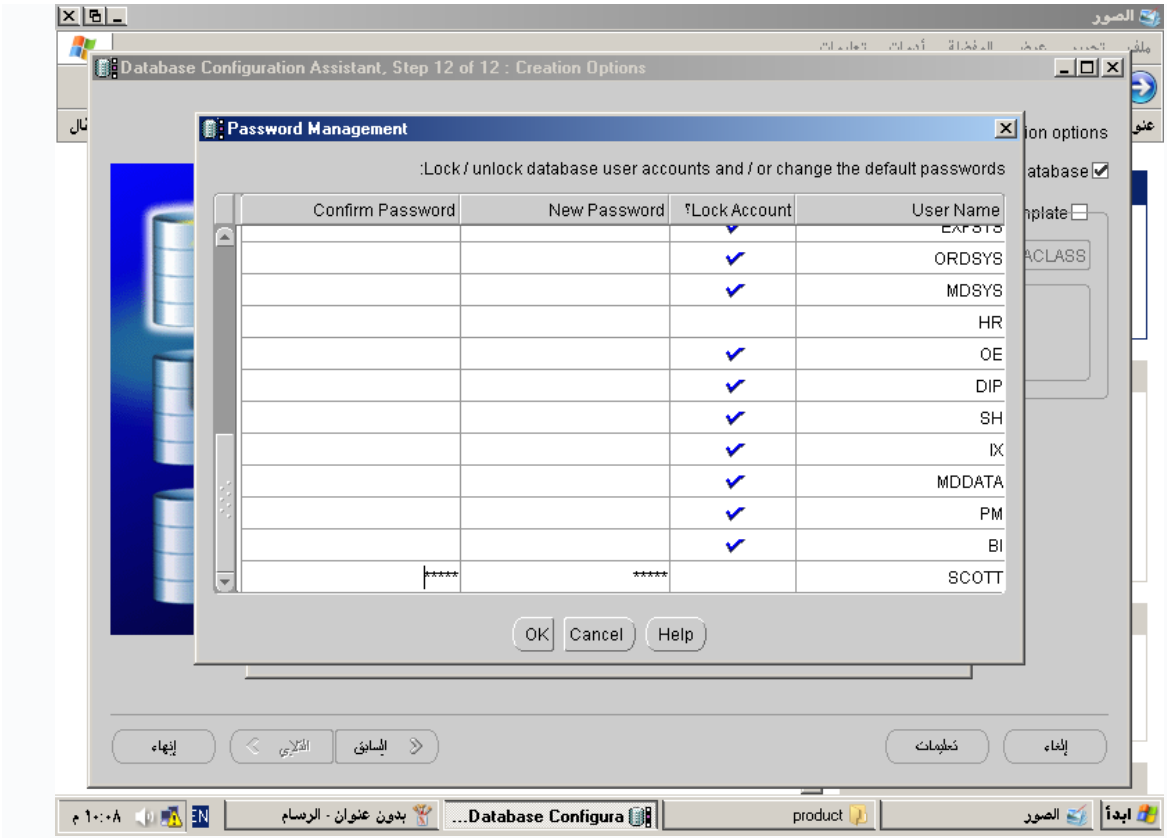


تظهر الواجهة التالية :

الغي اشارة صح على الحساب المراد فتحة وإذا ارت تعديل كلمة المرور اكتبها في المربعات الفارغة امام حساب المستخدم و في حالة تركها فارغة سوف يعتمد كلمة السر التي ادخلنها سابقاً وهي (b)



اضغط OK



و بعدها Exit

..كيف نحدد من هي القاعدة التي نتعامل معها عند تشغيل تطبيق مثلاً SQL plus

: نذهب إلى المسار التالي

C:\oracle\product\10.1.0\Db_1\NETWORK\ADMIN tnsnames.ora

: في جهازي tnsnames.ora سأضع لكم ما يحويه

```
# tnsnames.ora Network Configuration File:
C:\oracle\product\10.1.0\Db_1\network\admin\tnsnames.ora

# Generated by Oracle configuration tools.


DBACLASS =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = SICOWIN)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
```

```
    (SERVICE_NAME = DBACLASS.jzeera.com)
   )
  )

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = SICOWIN)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
```

بحيث تأخذ الشكل التالي :

```
 Connection_Host=
  (DESCRIPTION =
```

(ADDRESS = (PROTOCOL = TCP)(HOST = PC_NAME " اسم الجهاز الذي
" يحوي القاعدة)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME =GLOBAL_DATABASE_NAME)
)

... التي قمنا بإنشاءها DBCLASS و ORCL : كما تلاحظون فلدي قاعدتين
و SQL PLUSسلسة المضيف) أي عندما أريد أن أسجل الدخول على) connection Host ما يهمنا هو
سأكتب DBCLASS التعامل مع قاعدتنا

connection Host DBCLASS

ORCL: أما إذا أردت الدخول على القاعدة الثانية

connection Host ORCL

حذف قاعدة البيانات :

١- من قائمة إبداء – لوحة التحكم

٢- نختار الأداء و الصيانة

٣- نختار أدوات إدارية:

٤- نختار خدمات

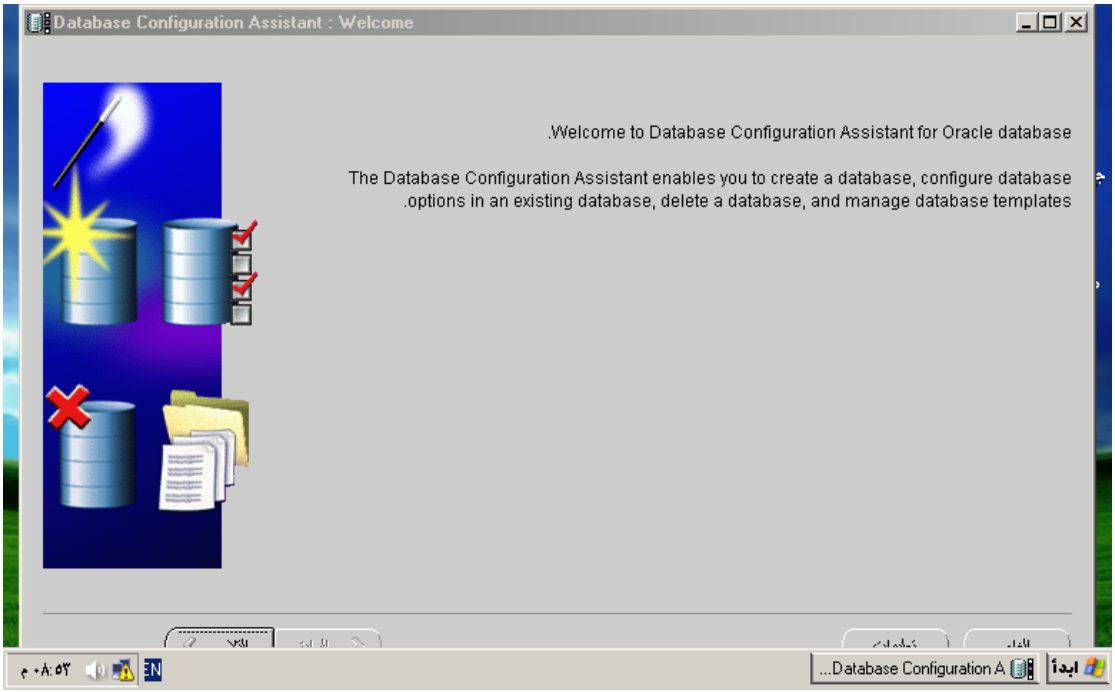
و

٥- نضغط بزر الماوس الايمن فوق OracleserviceORCL ونختار إيقاف

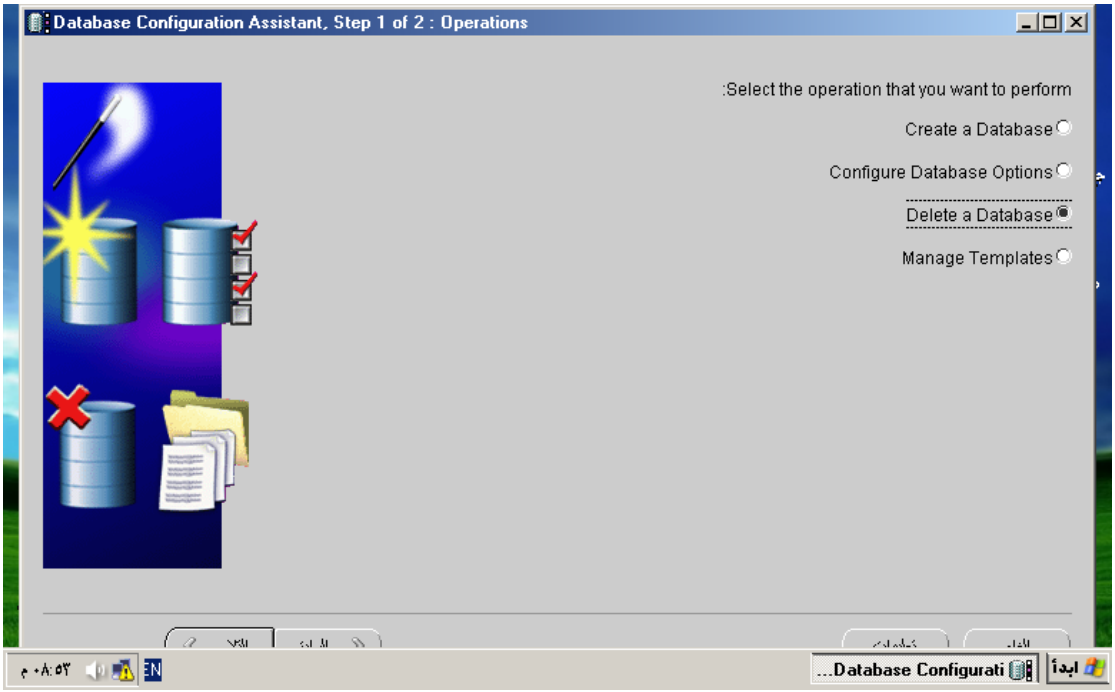
الان نبدأ عملية الحذف بعد لإيقاف خادم اوراكل.

١- من قائمة ابداء نختار

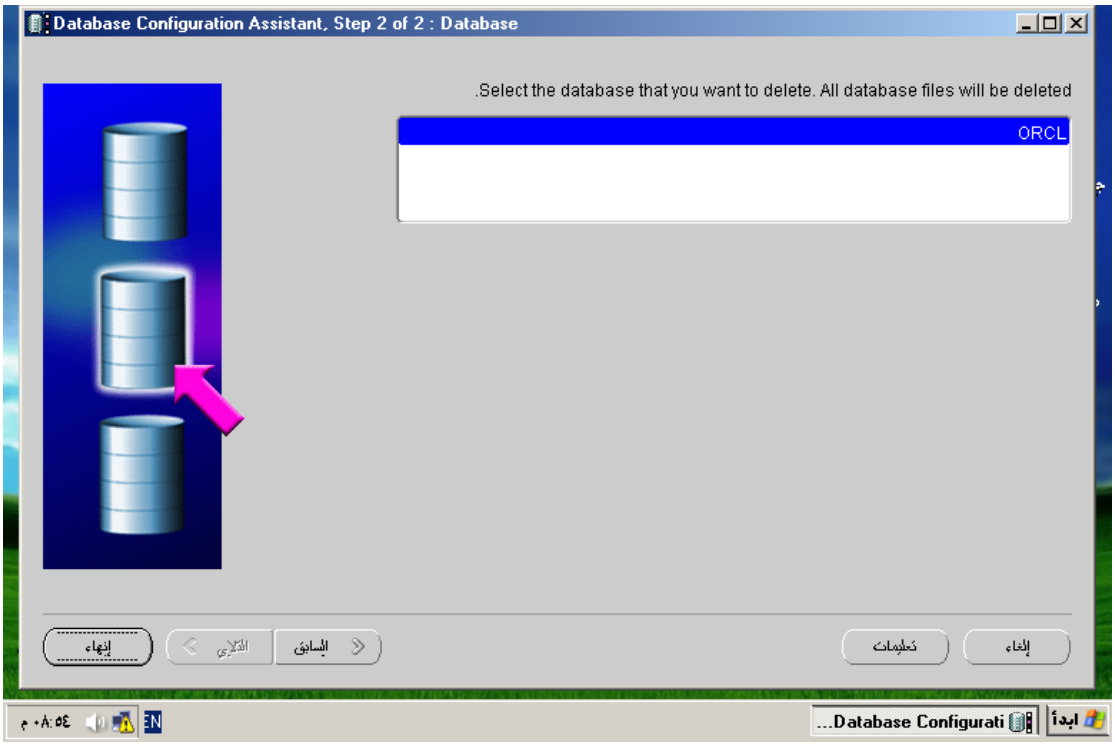
٢- نضغط التالي:



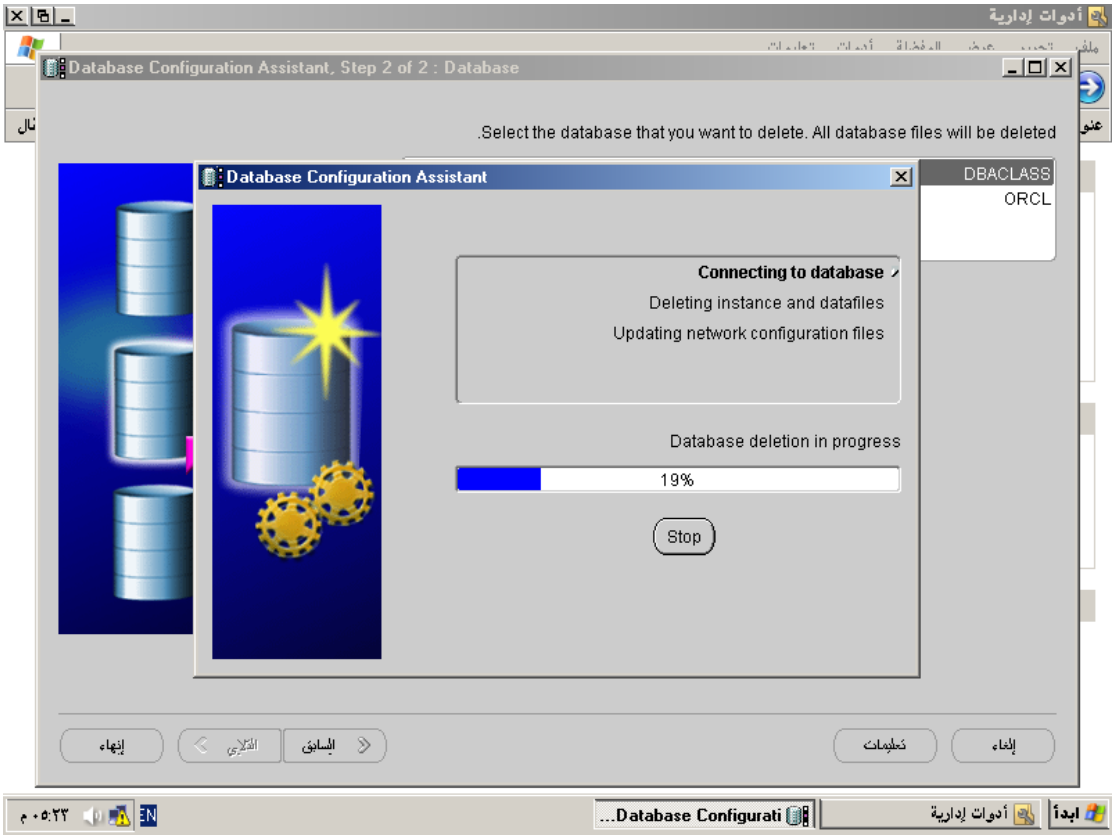
٣- نختار Delete database



٤- نحدد قاعدة البيانات المراد حذفها



٥- نضغط إنهاء للبدء في عملية الحذف:

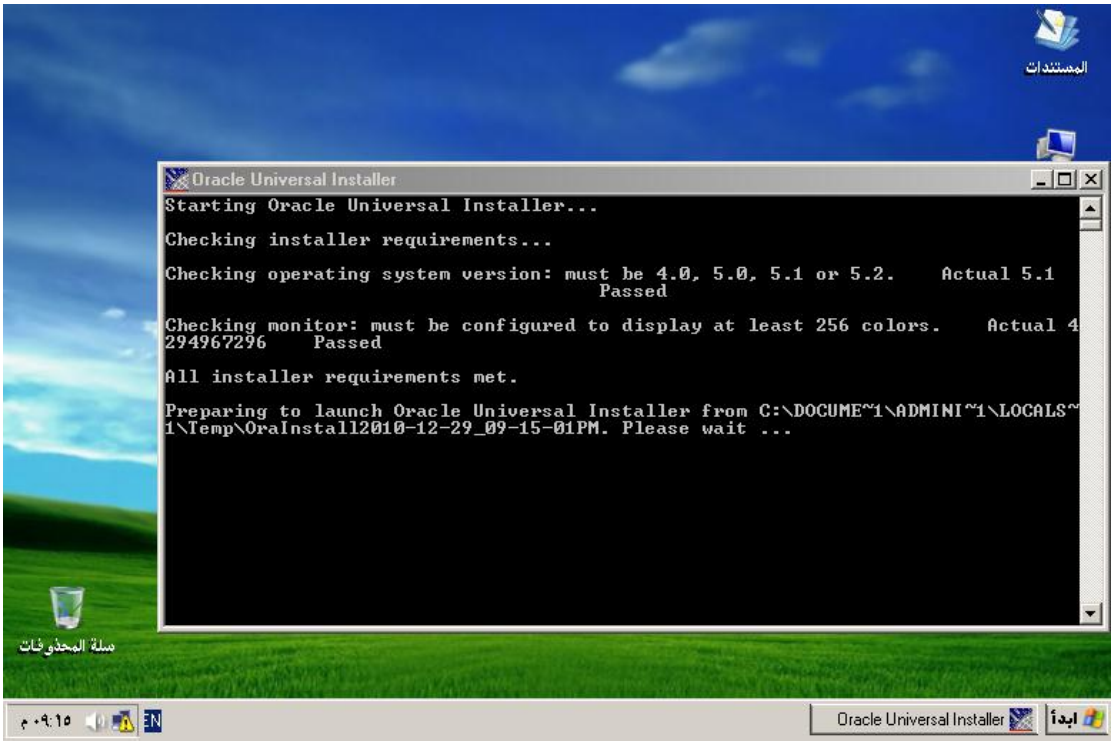

أدوات إدارية
Database Configuration Assistant, Step 2 of 2 : Database
.Select the database that you want to delete. All database files will be deleted
Database Configuration Assistant
DBACLASS
ORCL
Connecting to database
Deleting instance and datafiles
Updating network configuration files
Database deletion in progress
19%
Stop
إنهاء
التالي
السابق
تعليمات
إلغاء
...Database Configurati
أدوات إدارية
ابدأ

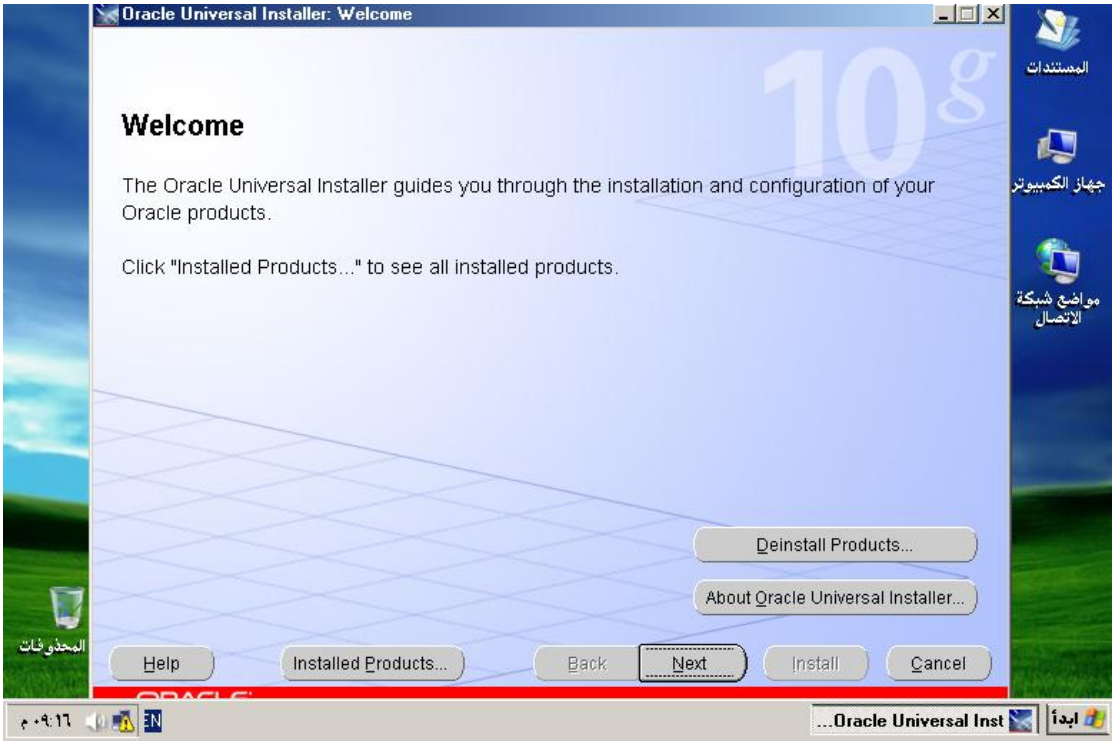
حذف اوراكل :

1- قم بإطفاء قاعدة البيانات بالكامل عن طريق أمر.SHUTDOWN IMMEDIATE

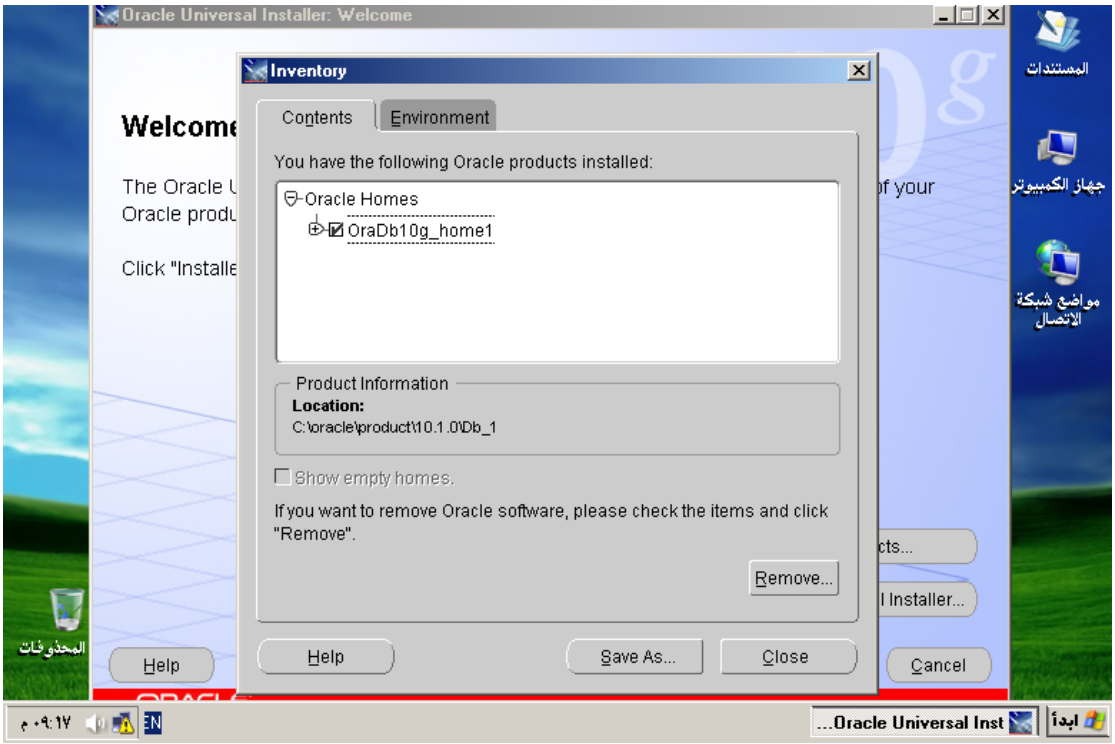
2. اذهب الى ال Services وقم بإيقاف اي Service تبدء بكلمه. Oracle
من قائمة ابداء – لوحة التحكم – أداء الصيانة – ادوات ادارية – الخدمات

3.قم بتشغيل برنامج Oracle Universal Installer وقم ب Deinstallبرامج اوراكل.

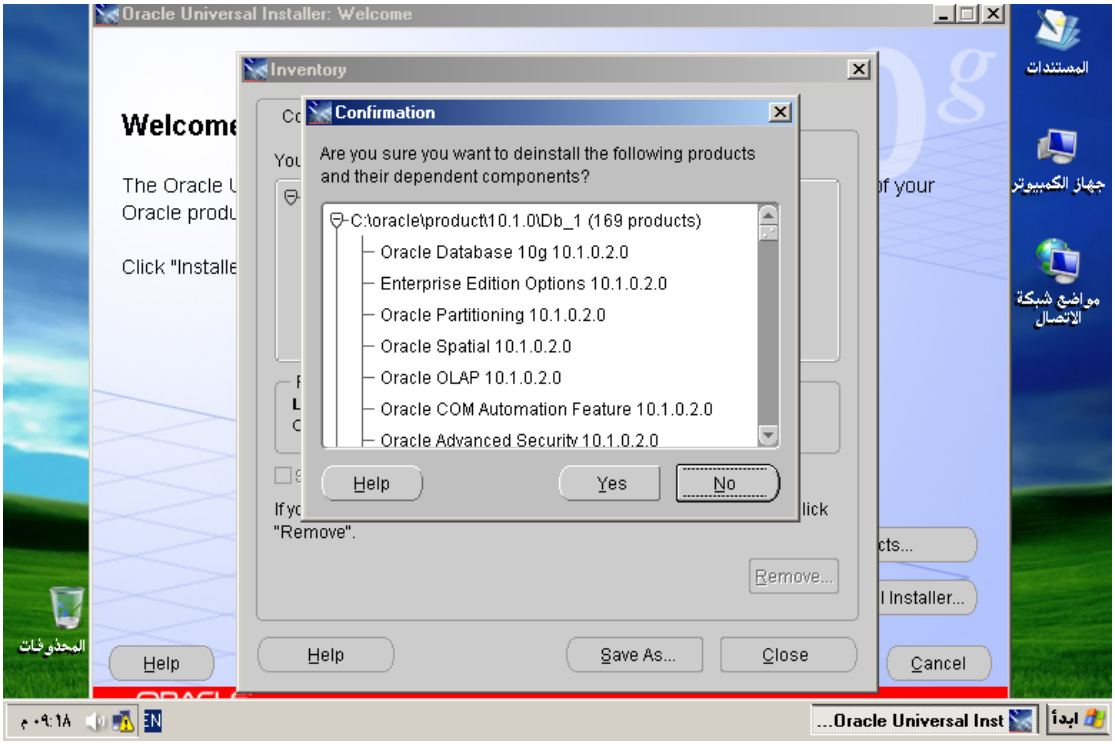




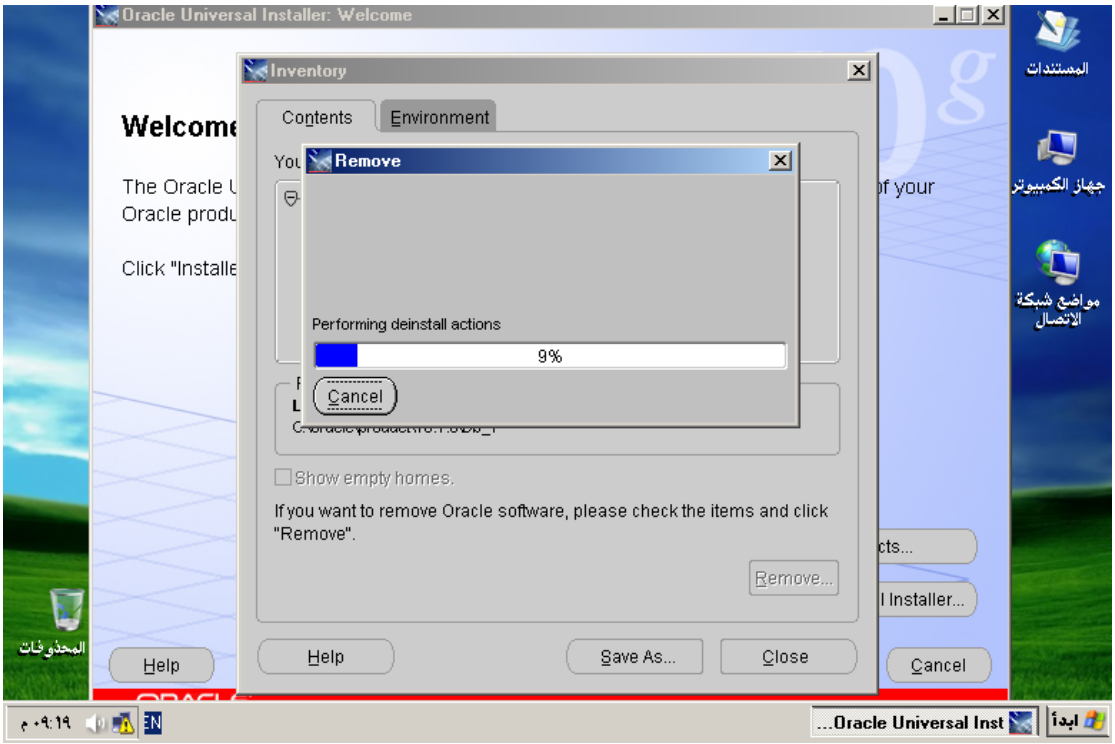
نضغط بDeinstall



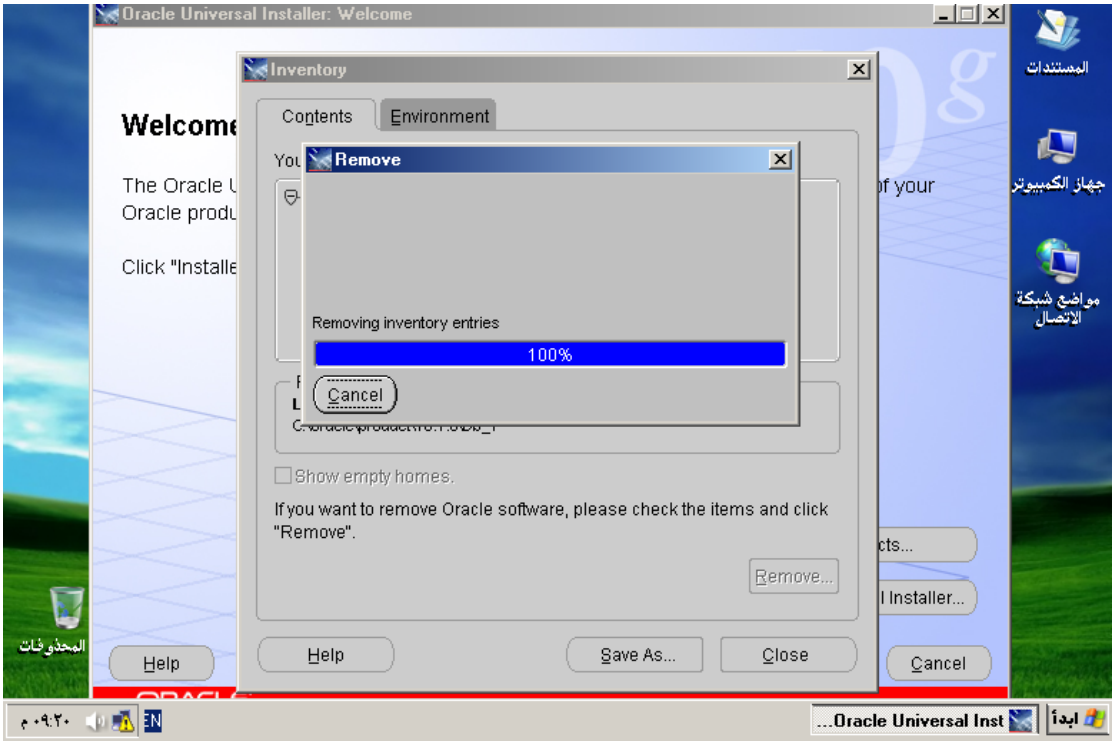
تظهر الواجهة السابقة نحدد قاعدة البيانات و نضغط زر Remove

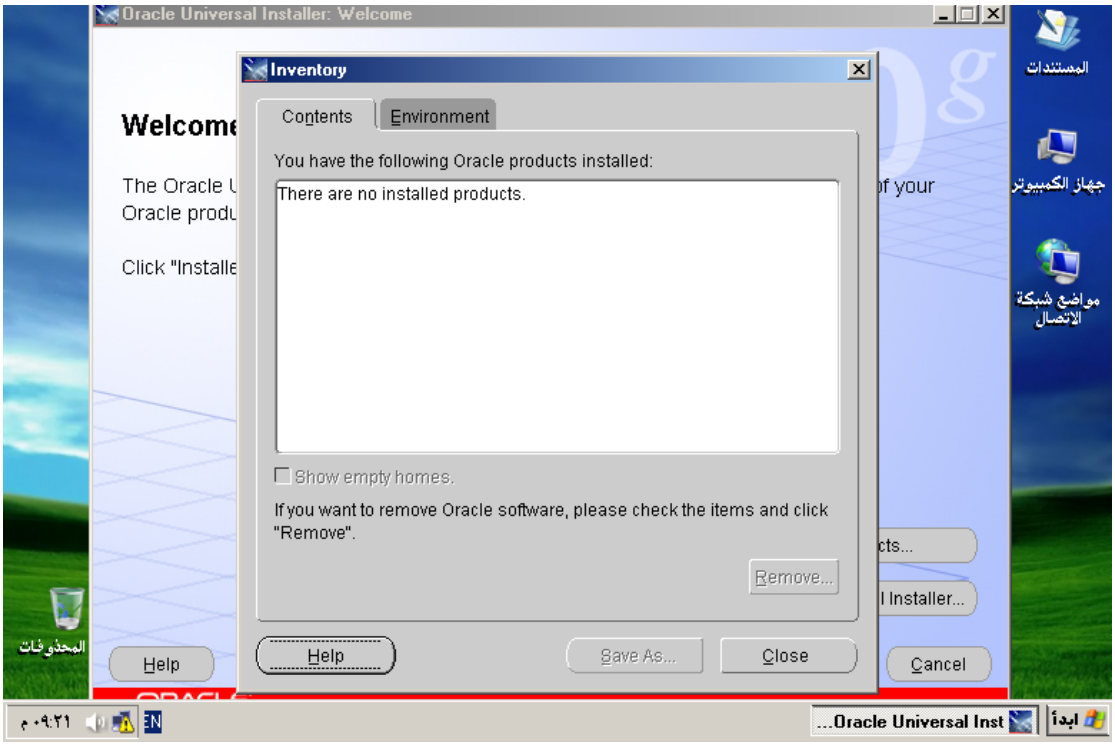


نضغط YES



وننتظر حتي الانتهاء من الحذف



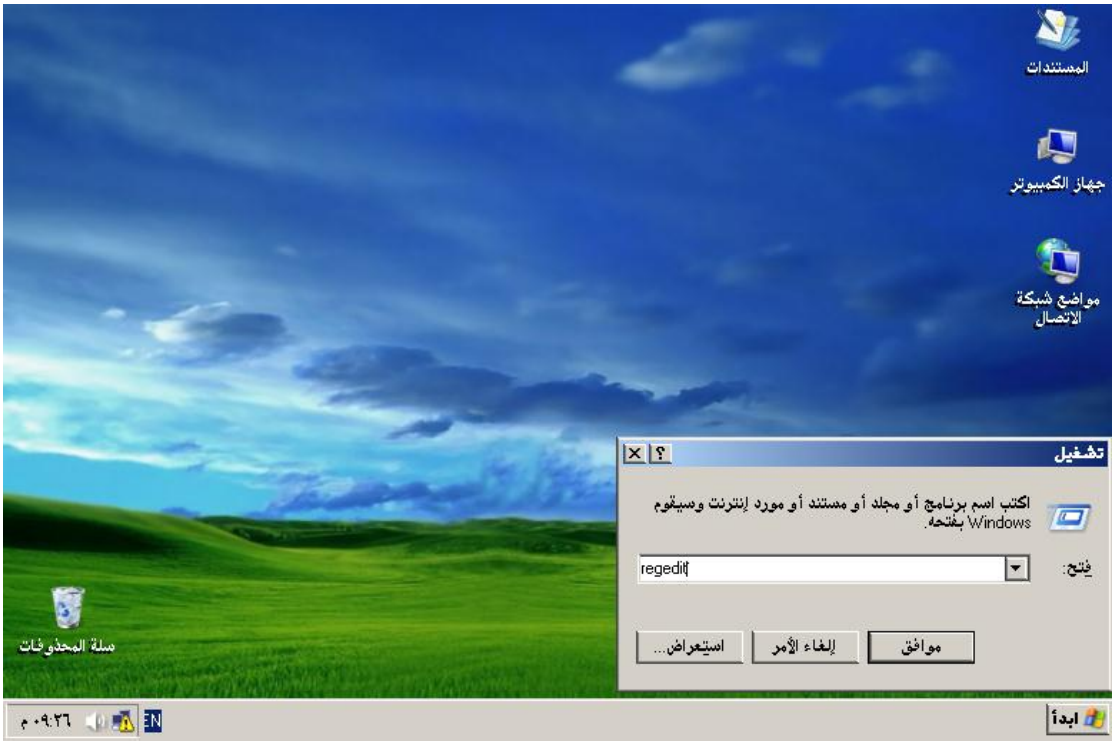


نضغط زر Close

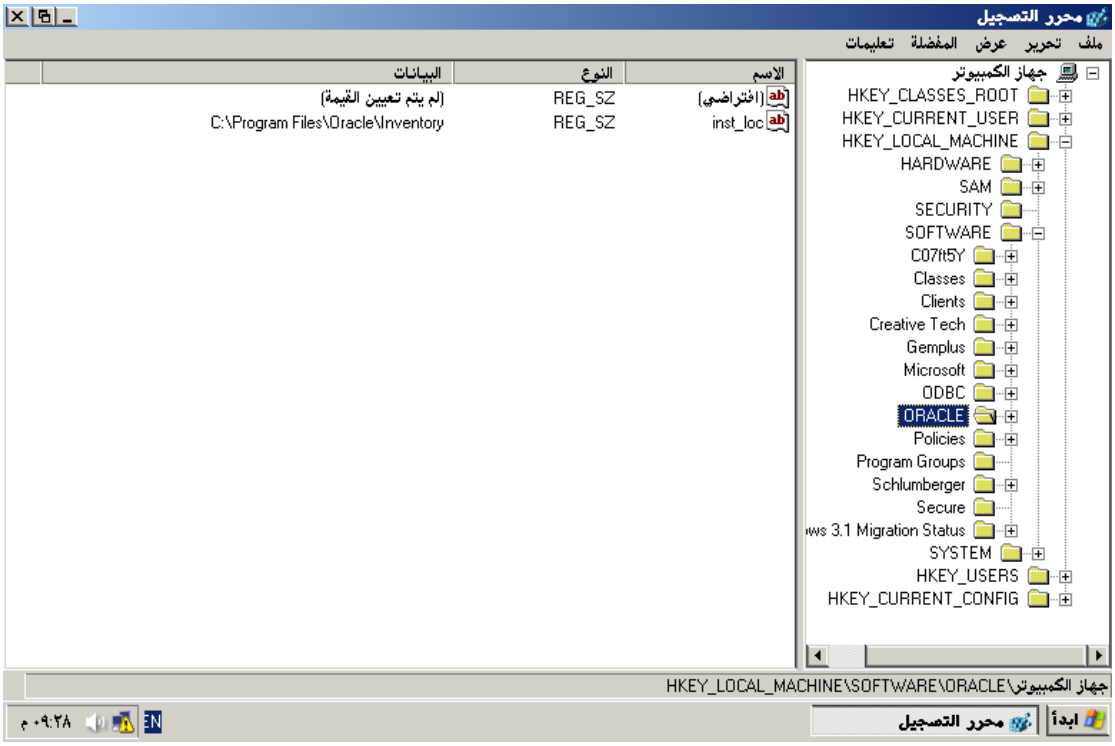
وبعدها Cancel

تنظيف الرجيستري-:

1. Select Start > Run, enter REGEDIT, and click OK.



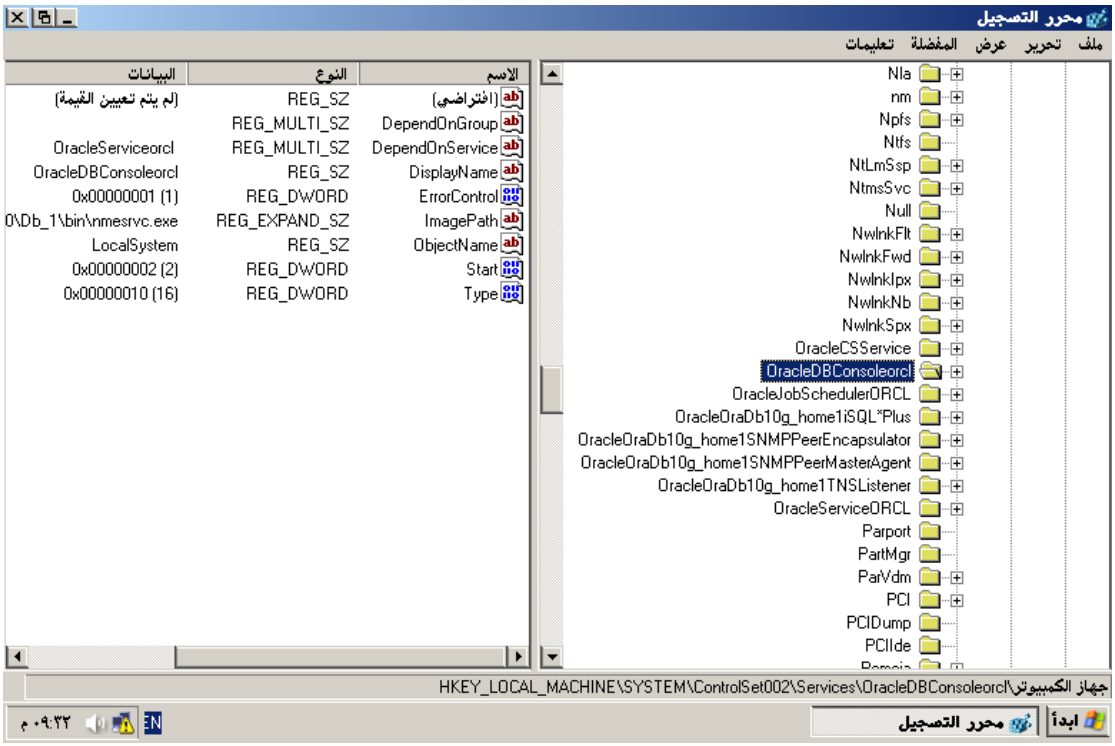
2.انتقل الى HKEY_LOCAL_MACHINE\SOFTWARE وقم بحذف ملف ORACLE بالكامل .



3.انتقل الى -:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

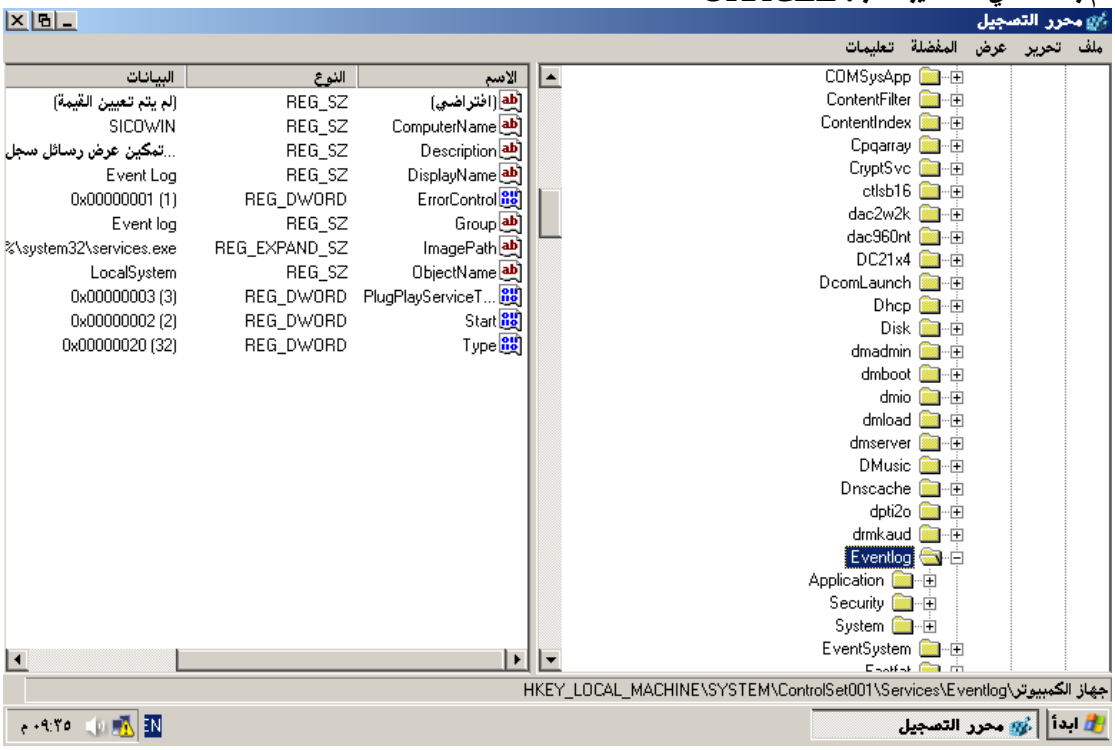
قم بحذف اي ملف يبدء ب. ORACLE



انتقل الى -:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog

قم بحذف اي ملف يبدء ب. ORACLE

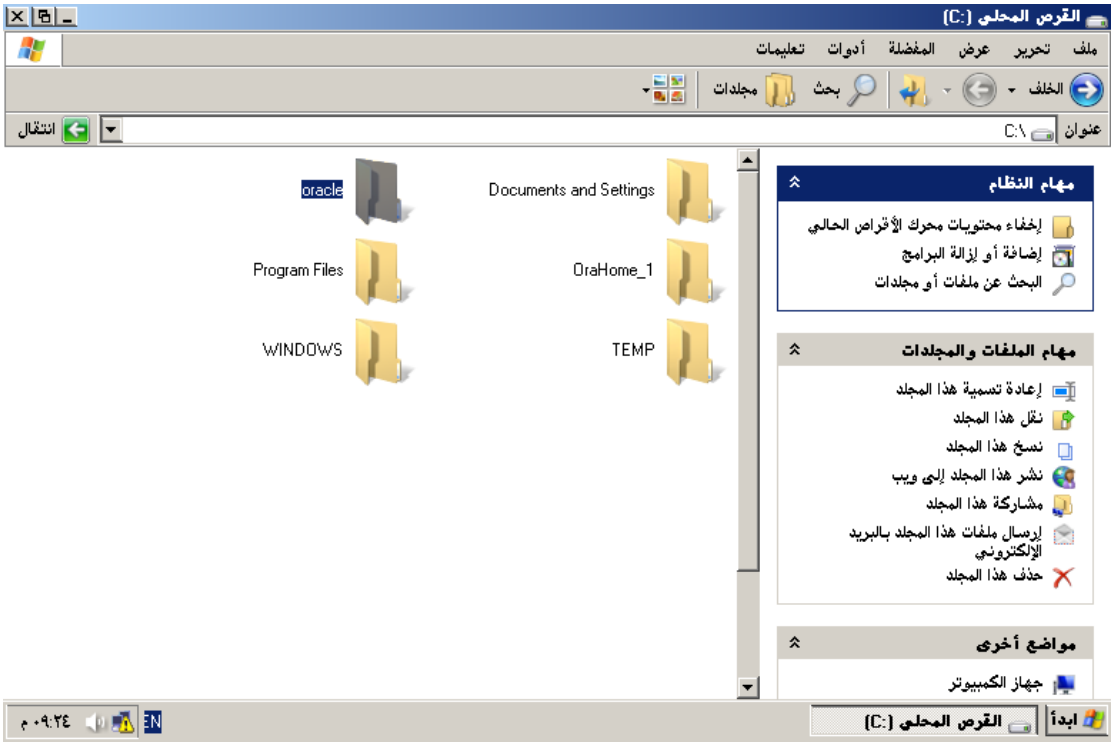


كرر العملية رقم ٣ و ٤ لأي ControlSet تجده بعد HKEY_LOCAL_MACHINE\SYSTEM فقد تجد ١او ٢

أو اكثر ملفات باسم ControlSet وامامها ارقام متسلسله تبدء من ٠٠٠ فقم بتكرار العملية رقم ٣ و٤ عليهم كلهم .
أعد تشغيل الجهاز وهكذا تكون قد حذفت الاوراكل بالكامل بإذن الله من جهازك والان تستطيع ان تنصب الاوراكل من جديد.

٥- أحذف ال " Directories الملفات " التي قمت بتنصيبها براميج اوراكل بها مثل , C:\Oracle ايضا قم بحذف الملف التالي

C:\PROGRAM FILES\ORACLE .



ما معنى إغلاق القاعدة أو تشغيلها ؟

في المحاضرات السابقة تحدثنا عن أن الأوراكل سيرفر الذي نتعامل معه مؤلف من قسمين:Oracle instance و القاعدة .. Oracle Database .
رأينا أن Instance هو فعلياً برنامج Software يستجيب لطلباتنا و يقوم بتنفيذها و يعبد إلينا النتائج إضافة إلى وظائفه الأخرى في الحفاظ على القاعدة .... إلخ ..
أما القاعدة فقد وجدنا أنها عبارة عن ملفات فيزيائية تفتح و تدار من قبل .. Oracle instance

ما معنى إغلاق القاعدة ؟
المعنى أنك أولاً تغلق ملفات القاعدة المفتوحة ثم تقوم بإيقاف... Oracle instance
ماذا عن تشغيلها ؟
يعني التشغيل أنك أولاً تقوم بتشغيل Oracle Instance الذي يقوم بفتح ملفات القاعدة...

سجل الدخول بالشكل التالي :

```
sys/pass as sysdba
```

لعمليات التي تحدث عند تشغيل Instance

تتوزع العمليات بين ثلاثة مراحل
:

المرحلة الأولى : nomount
-1 يقوم أولاً بقراءة ملف البارامترات .. Parameters File هذه البارامترات على سبيل المثال تذكر ما سم القاعدة التي ستعمل معه instance و أين هو Control fileالخاص بالقاعدة ما هو حجم الذاكرة التي سيحجزها .. إلخ
وهذه المتغيرات موجوده في إحدى الملفات التالية:
اولاً spfileSID.ora.

*** اذا لم يجده بيحث عن spfile.ora.**

*** إذا لم يجده يبحث عن initSID.ora.**

في المسار التالي :

C:\oracle\product\10.1.0\Db_1\database

-2 يقوم بحجز الذاكرة .
-3 يشغل .. Background process
-4يفتح ملف alertSID.log و هو أهم ملف يعتمد عليه مدير القاعدة لكي يكتشف الأعطال و المشاكل الذي تحدث للقاعدة إضافة إلى معلومات مهمة جدا سنراها لاحقاً ..... يوجد هذا الملف في المسار التالي

```
C:\oracle\product\10.1.0\admin\ORCL\bdump\ alert_ORCL.log
```

المرحلة الثانية : Mount :

يقوم بقراءة ملف Control File لكي يحدد أين توجد ملفات البيانات Data filesو ملفات الإرجاع Redo Files .... إلخ و لكنه لا يتأكد من وجود تلك الملفات ..
هذه المرحلة مهمة جداً لأن عملية أخذ النسخ الاحتياطية و إصلاح ملفات البيانات و ملفات الإرجاع يتم في هذه المرحلة ... كما سنرى لاحقاً .

المرحلة الثالثة : Open :

يقوم بفتح ملفات البيانات Data files و ملفات الإرجاع Redo log في هذه المرحلة .. نجاح هذه المرحلة يعني أن القاعدة قد أصبحت جاهزة تماماً لإستقبال الطلبات ...

يمكن لمدير القاعدة أن يحدد أي مرحلة يريد قاعدته أن تعمل بها و ذلك تبعاً للعملية التي يريد تنفيذها و سأذكر أمثلة فقط :
-1إذا كانت هناك مشكلة في Control file فيقوم بتشغيل القاعدة في مرحلة .. nomount
-2إذا كانت هنالك مشكلة في ملفات البيانات أو ملفات الإرجاع أو أراد أخذ نسخة إحتياطية للقاعدة فإنه يقوم بتشغيل القاعدة في وضع .. Mount

يتم أمر التشغيل كالتالي :
-1تسجيل الدخول أولاً بالحساب SYS AS SYSDBA على برنامج . SQLPLUS
-2تنفيذ الأمر :

```
startup Level_name;
```

لتنفيد إي مرحلة يجب إغلاق قاعدة البيانات بتنفيذ الامر:

```
shutdown immediate;
```

لشتغيل القاعدة مثلاً في وضع : No Mount

```
startup nomount;
```

لتشغيل القاعدة في وضع : Mount

```
startup mount;
```

لتشغيل القاعدة لتصبح في الوضع الطبيعي :

```
startup;
```

ما ذا لو أردنا الإنتقال بين المراحل ؟

-1نشغل القاعدة بالوضع : nomount

```
startup nomount;
```

-2للإنتقال إلى الوضع : mount

```
alter database mount;
```

-3للإنتقال إلى الوضع النهائي :

```
alter database open;
```

ثانياً: إغلاق قاعدة البيانات

يتم إغلاق القاعدة بعكس العمليات التي تنفذ ضمن عملية التشغيل :

1 - يتم إغلاق ملفات البيانات Data files ومسجلات الإرجاع.. Redo Files
2-يتم إغلاق ملف .. Control file
3- يتم تحرير الذاكرة و إيقاف Background Process و إنهاء instance
قبل أن أبدأ بشرح الإغلاق دعوني أتحدث عن موضوع مهم :
المناقلات : Transactions
تعني المناقلة مجموعة التعديلات بواسطة تعليمات DML التي يجريها المستخدم ضمن جلسته الحالية ... و هذه التعديلات تكون مؤقتة يمكن التراجع عنها أو تثبيتها بحسب رغبة المستخدم ..
يمكنك تجربة المناقلة ببساطة :
نفذ بعض تعليمات insert أو update أو delete على جدول ما ثم نفذ التعليمة التالية :

```
rollback;
```

فتجد أن جميع التعديلات التي أدخلتها قد ولت أدراج الرياح و كأنك لم تفعل شيئاً .. إذاً ما الذي فعلته التعليمة Rollback؟؟
ماقامت به التعليمة هي مجرد التراجع عن المناقلة التي قمت بها و أعادت كل شيء إلى طبيعتها ...
حسناً : ماذا لو أردت تثبيت التعديلات لتصبح دائمة ؟

```
commit;
```

لنعد إلى إغلاق القاعدة :
لإغلاق القاعدة تحتاج إلى تنفيذ التعليمة التالية :

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT ]
```

لنبدأ بالشرح :

١- Normal

و هو الخيار الافتراضي حيث ينتظر النظام جميع المستخدمين لكي ينتهوا من عملهم مع القاعدة و يقوموا بإغلاق اتصالاتهم (تسجيل الخروج من SQLPLUS أو إغلاقه أو حتى إغلاق البرامج التي تتعامل مع الأوراكل) ... هذا الأمر يحفظ جميع المناقلات و لن يسبب لك أي مشكلة عند تشغيل القاعدة مرة أخرى ..

مثال : يوضح ذلك ؟
-1شغل SQLPLUS و سجل الدخول مثلاً بالحساب .. Scott
-2شغل نسخة أخرى من SQLPLUS و سجل الدخول بالحساب... SYS AS SYSDBA و نفذ التعليمة :

```
shutdown normal;
```

فتجد أنه ينتظر ...
-3أغلق نسخة SQLPLUS التي استعملتها مع Scott فستجد أنه بدأ بإغلاق القاعدة عند النسخة الثانية ....

2- Transactional

هنا ينتظر النظام المستخدمين الذي أجروا تعديلات على القاعدة و لم يثبتوا تعديلاتهم بعد ، أما أولئك المستخدمين الذين لم يجروا أي تعديلات أو ثبتوا تعديلاتهم فإنه مباشرة يقوم بقطع إتصالاتهم مع القاعدة ..

مثال :

-1افتح نافذة SQLPLUS و سجل الدخول بالحساب Scott ثم نفذ تعليمة update أو delete أو insert على أي جدول تريد و لا تقم بعملية التثبيت...

-2افتح نافذة SQLPLUS جديدة و سجل الدخول بالحساب SYS AS SYSDBA ثم حاول إغلاق القاعدة بالتعليمة التالية :

```
shutdown transactional;
```

ستجد أنه ينتظر ...

-3نفذ تعليمة تثبيت التعديلات في نافذة : scott

```
commit;
```

ستجد إن عملية الإغلاق في نافذة SYS قد بدأت و سوف تنتهي بإغلاق القاعدة ...

-4 حاول تنفيذ أي تعليمة في نافذة scott فإنها ستفشل و سيعطيك إحدى الرسائل Oracle not available :أو Shutdown in progress .....

Immediate -3

هنا لن ينتظر أحد ، بل سيقوم بالتراجع عن التغييرات لدى المستخدمين الذي لم يثبتوا تعديلاتهم بعد ثم سيقطع الإتصال مع الجميع و يغلق القاعدة ....

نفذ السيناريو التالي :

-1افتح نافذة SQLPLUS و سجل الدخول بالحساب Scott ثم نفذ تعليمة update مثلاً:

```
update emp set sal=0 ;
```

لا تقم بالتثبيت ..

- -2افتح نافذة SQLPLUS جديدة و سجل الدخول بالحساب SYS AS SYSDBA ثم حاول إغلاق القاعدة بالتعليمة التالية :

```
shutdown transactional;
```

ستجد أنه لن ينتظر و سيقوم بإغلاق القاعدة مباشرة ..

-3أعد تشغيل القاعدة:

```
startup;
```

-4سجل الدخول مرة أخرى في نافذة scott ثم نفذ تعليمة : select

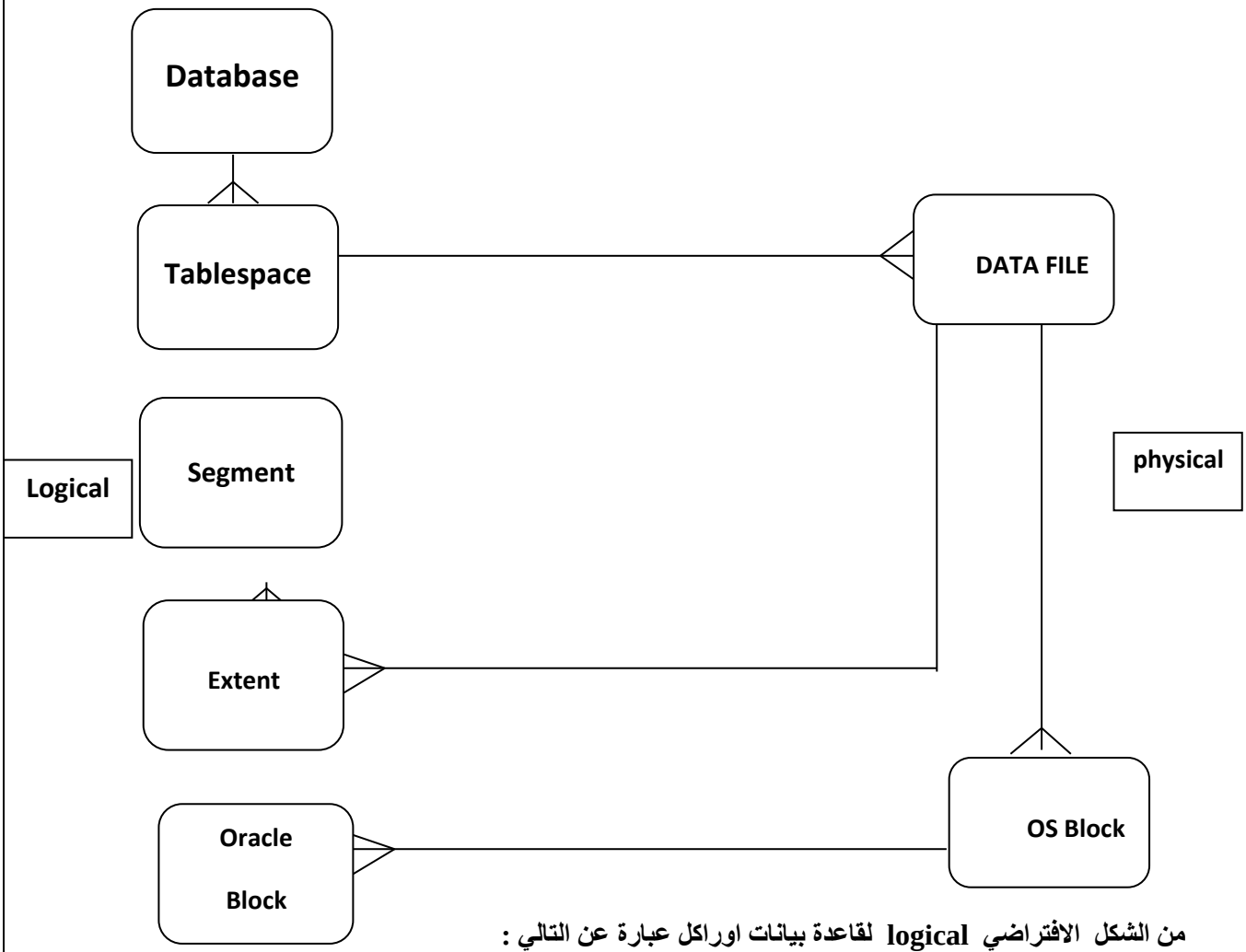
```
select sal from emp;
```

ستجد أن الأمور قد عادت إلى طبيعتها و أن التعديل الذي جعلنا فيه sal=0 قد تم التراجع عنه ...

4- ABORT

و هو أسوء خيار قد تلجأ إليه ، فهو يقوم بعملية إغلاق قسري للقاعدة دون المرور بالخطوات التي تحدثنا عنها سابقاً ، هذه العملية ستؤدي إلى آثار سلبية يتم إصلاحها بواسطة عملية نسميها Instance Recovery عند تشغيل القاعدة مرة أخرى ، لذلك لا تلجأ إليه إلا إذا كنت مضطراً إلى هذه العملية....

مكونات قاعدة البيانات :



من الشكل الافتراضي logical لقاعدة بيانات اوراكل عبارة عن التالي :

1- Database : مكونة من واحد أو أكثر من المساحات الجدولية TableSpace

2- Tablespace : مكونة من واحد أو أكثر من القطاعات Segment علي المستوى المنطقي (ومكونة من ملف واحد أو أكثر على المستوى الفيزيائي).

3- Segment : أن كل قطاع سواء كان قطاع table أو index مكون من امتدادات Extents

4- Extents : عبارة عن مجموعات متجاورة من البلوكات وهى اصغر وحدة من وحدات التخزين تستخدم بواسطة أوراكل

ملفات Data Files

- تعتبر أهم مكون من مكونات قاعدة البيانات حيث تحوى جميع Index table والتى تخزن بها البيانات على هيئة صفوف ٠ تلك الملفات لا يمكن قراءتها مباشرة بواسطة اية وسيلة من محررات النصوص العادية وانما يتم التعامل معها بواسطة اوراكل عن طريق تنفيذ جمل استعلام معينة ٠ان تلك المعلومة هامة للغاية فاننا لا نملك السيطرة على كيفية تخزين data فى الملفات المادية فهذا دور اوراكل ولكننا فقط نستطيع التعامل مع البيانات من خلال تنفيذ جمل SQL على هيكل افتراضى مكون من المساحات الجدولية TableSpaces والتى تنشا فيها الجداول Tables وفهارس Indexes كما يتضح من الشكل التالي

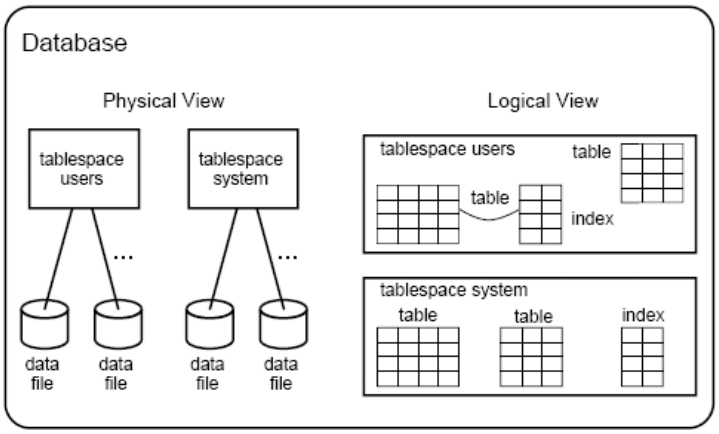


Figure 8.3. Physical versus logical data structures.

١. **المساحة الجدولية TableSpace** :

هي التمثيل الحقيقي للملفات الفيزيائية في الذاكرة .

- يمكن تشبيها بوعاء كبير حاوي لجميع القطاعات فكل قطاع Segment يجب ان ينتمي إلى مساحة جدوليه خاصة به التي تحوى الامتدادات Extents والتي تحوى البلوكات Block ٠

أنواع TableSpace :

١- System Tablespace :

a. وهي المساحة الافتراضية التي تنشأ عند إنشاء قاعدة البيانات

b. لا يمكن حذفها ولكن يمكن زيادة حجمها .

c. يخزن فيها Data dictionary

٢- NON-System:

a. وهي المساحة التي ينشئها المستخدم

نوع الفضاءات الجدولية من حيث التخزين :

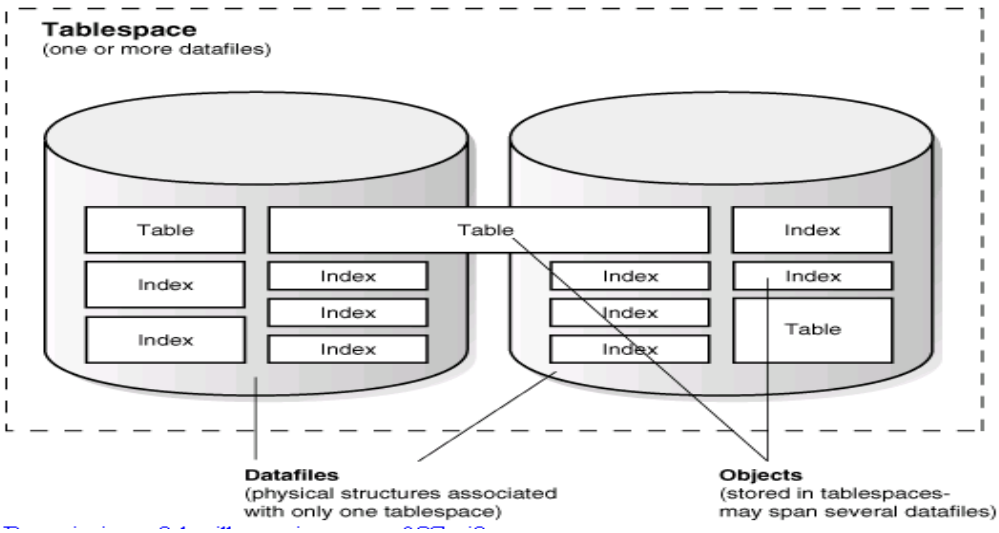
1- permanent تستخدم لتخزين البيانات بشكل دائم ، و هذا ما نستخدمه معظم الأوقات ..

2- temporary يستخدم لتخزين البيانات المؤقتة و نستخدمه إذا اردنا استعمال مثلاً عمليات ترتيب على البيانات قبل استخدامها ...

3- undo مخصصة لتخزين بيانات التعديل المؤقت (كانت هذه مهمة Rollback segments في الإصدار ٨ i و هي التعديلات التي يجريها المستخدم التي تحدثنا عنها عند شرح مفهوم المناقلة . Transaction)

لشكل التالي يوضح العلاقة بين المساحات الجدولية وملفات Data File و أنواع الكائنات التي تحتويها المساحة الجدولية

Figure 3-1 Datafiles and Tablespaces



٢- القطاع Segment:

و هي اسماء الأغراض التي نقوم بإنشائها داخل القاعدة : مثال الجدول emp هو segment و الفهرس indexهو segment ... إلخ

أو :

يمكن تمثيل القطاع Segment بالوعاء الذى يشغل حيزا معينا لتخزين البيانات ٠ وكما اننا يمكننا نخزن ماءا في وعاء معين فيكون وعاء يحتوى على الماء وربما احتجنا اخر لتخزين سائلا اخر فيه فيكون ايضا وعاء ا لتخزين الزيت مثلا كذلك كل كائن يتطلب تخزينه في قاعدة البيانات الوعاء الخاص به Table تكون مكونا من العديد الامتدادات القطاعية فيكون لدينا ما يسمى Table Sgment والفهارس تحتاج الى امتدادات مختلفة اخرى فتمسى امتدادها Index Sgments ومعلومات التراجع تسجل على امتدادات تسمى Rollback Sgments وهكذا ٠

٣- الامتداد Extents

يتم تقسيم ملفات البيانات إلى مساحات صغيرة تدعى كل مساحة .. Extent و سميت إمتدادات لأنه في العادة يوجد حد إبتدائي لحجم ملف البيانات ماذا لو وصلنا إلى هذا الحد و أردنا إضافة بيانات جديدة ؟؟ عندها سيتم حجز مساحة فيزيائية إضافية للملف بمقدار حجم Extentالمحدد بالتالي يصبح هناك مساحة لتخزين البيانات الجديدة

الآن عندما نريد إضافة سجل جديد إلى الجدول emp ما الذي سيقوم Oracle بفعله ؟؟ سيقوم بالذهاب إلى الفضاء الجدولي الذي قمنا بتخزين الجدول داخله و منه سيحصل على اسم ملف البيانات و عندها سيتحقق هل يوجد أي مساحة فارغة في أي Extent ؟؟ أذا لم يوجد فإنه يقوم بحجز Extent جديد و يقوم بالتخزين داخله ...

إذاً ا عملية التأكد من وجود Extent فارغ و حجز مساحات جديدة أو إزالة مساحات فارغة نسميها إدارة الإمتدادات .. Extent Management

إن هذه الإدارة تعتمد على بنية تبين أي إمتداد Extentيحوي على مساحة فارغة بالتالي هذه البنية إما أن تكون مخزنة داخل الفضاء الجدولي نفسه و هذا ما نسميه

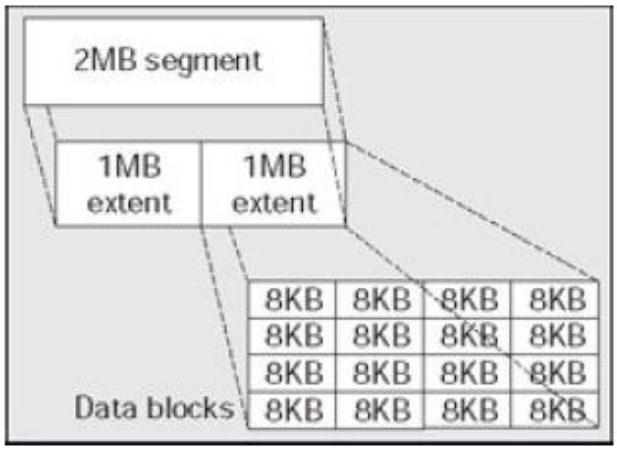
Locally Managed Tablespace أو تكون مخزنة داخل قاموس البياناتData Dictionary

(و هو مجموعة من الجداول و المشاهد التي يملكها المستخدم sysالتي توصف القاعدة من مستخدميها و صلاحياتهم إلى جداولها و فهارسها و إجراءاتها .... إلخ) و هنا ما يسمى

Dictionary Managed tablespace

٤- البلوك Block

الامتدادات تتكون من بلوكات Bloack وهى اصغر وحدة تخزينية يمكن حجزها وتخصيصها فى اوراكل ٠ ان البلوكات هى ما سوف تخزن به اوراكل الداتا فعليا Physical فى ملفات على نظام التشغيل ٠ ان البلوك هو اصغر وحدة يمكن قراءتها او كتابتها من الديسك ويلاحظ ان البلوك فى اوراكل يختلف عن البلوك فى نظام التشغيل مثلا فهو يتراوح بين (٢او ٤ او ٨ او حتى ١٦ كيلو بايت او ٣٢ كيلو بايت)



الصيغة العامة للانشاء مساحة جدولية LOCAL:

```
CREATE  TABLESPACE  name_tablaspace   SIZE n K|M
DATAFILE  ['file_space']
[AUTOEXTEND
              OFF
ON     NEXT n k|m   MAXSIZE  UNLIMITED | n K|M}
[LOGGING |NOLOGGING]
[PERMANENT]
[EXTENT MANAGEMENT
                 LOCAL
                 AUTOALLOCATE
                       UNIFORM    SIZE  n k|M  ]
[ONLINE|OFFLINE]
```

شرح الصيغة العامة:

1- tablespace : اسم المساحة الجدولية التي نريد إنشاءها

2- DATAFILE : يحدد الملف أو الملفات التي تؤلف المساحة الجدولية .

3- AUTOEXTEND :يعطل أو يفعل التمدد التلقائي لملفات البيانات

4- OFF : تعطيل

5- ON تفعيل

6- مساحة القرص التي ستخصص لملف عند طلب المزيد من الامتدادات

7- مساحة القرص الأعظميه التي يسمح بتخصيصها لملف البيانات

8- UNLIMITED لايضع حد لمساحة القرص المخصصة لملف البيانات .

9- LOGGING : إمكانية تسجيل كافة التغيرات التي تطرأ علي Table و Index الموجودة في هذه المساحة في الملف REDO LOG FILE

10- NOLOGGING : عدم إمكانية تسجيل كافة التغيرات التي تطرأ علي Table و Index الموجودة في هذه المساحة في الملف REDO LOG FILE

11- ONLINE : يجعل المساحة الجدولية متاحة مباشرة بعد إنشائها للمستخدمين الذين قد منحوا حق الوصول إليها

12- OFFLINE : يجعل المساحة الجدولية غير متاحة مباشرة بعد إنشائها

13- EXTENT MANAGEMENT : يحدد طريقة إدارة الامتدادات في المساحة الجد ولية .

14- UNIFORM : إمكانية جعل جميع الامتدادات بنفس الحجم .

15- AUTOALLOCATE : هو جعل أوراكل هو من يحدد أحجام الامتدادات في المساحة الجد ولية أوتوماتيكيا

تطبق عملي :

١- إنشاء مساحة جدوليه دائمة :

```
CREATE TABLESPACE  TESTm  DATAFILE c:\TEST\TEST11.DBF' SIZE 10M NOLOGGING;
```

تخصيصها للمستخدم اسمه (jaz) باستخدام ALTER USER اذا كان المستخدم موجود من قبل :

```
alter user jaz

default tablespace TEST

quota 5m on test;
```

فإن أي كان سوف ينشئه المستخدم jaz سوف يشاء في هذه المساحة التخزينية مثلاً:

```
create table mytab (num number);

Insert into mytab values(1);
```

للاستعلام عن المساحات الجدولية الموجودة في اوراكل

```
select *  from DBA_TABLESPACES
```

٢- زيادة حجم الTablespaces

الطريقة الاولى إضافة Data File جديد للTablespace.

```
alter tablespace TEST

add datafile 'c:\TEST\pp.DBF' SIZE 200M;
```

الطريقة الثانية زيادة حجم الData File الموجودة.

```
ALTER  DATABASE  DATAFILE 'c:\TEST\TEST1.DBF' RESIZE 12M
```

٣- تحويل الTablespace إلى الوضع Read Only

```
ALTER TABLESPACE TEST   READ  ONLY;
```

في هذه حالة لا يمكن إجراء أي تعديل على الكائنات الموجودة في هذه المساحة التخزينية :

```
update mytab set num=3;
```

عند تنفيذ التعليمة السابقة تظهر رسالة خطاء توضح أنة لا يمكن التعديل و إنما مسموح لك القراءة فقط.

٤- تحويل الTablespace الى الوضع Read Write:

```
ALTER TABLESPACE TEST READ WRITE;
```

٥-تحويل الTablespace الى الوضع Offline:

```
ALTER TABLESPACE TEST OFFLINE;
```

٦- نحويل الTablespace الى الوضع Online:

```
ALTER TABLESPACE TEST ONLINE;
```

٧- لإعادة تسمية الTablespace :

```
ALTER TABLESPACE TEST  RENAME  TO  EXAM;
```

و نعيدها للنفس الاسم ننفذ:

```
ALTER TABLESPACE  EXAM  RENAME  TO  TEST  ;
```

٨-نقل وإعادة تسمية الملفات MOVING DATAFILES

يمكن عمل ذلك باستخدام إحدى الطريقتين الآتية:

اولاً: باستخدام ALTER TABLESPACE

طريقة تنفيذ إعادة تسمية او نقل DATAFILE:

1- يجب إغلاق offline tablespace

٢- تنفيذ التغيير على مستوى Operating System

بنسخ الملف الي مكان اخر او إعادة تسميتة وبعدها تنفيذ هذه التغيرات على مستوى قاعدة البيانات بتعليمة التالية:

```
ALTER TABLESPACE TEST

RENAME

DATAFILE c:\TEST\pp.DBF'

TO c:\TEST\yy.DBF';
```

٣- تحويل online TABLESPACE
٤- يمكن حذف DATAFILE القديم
للنقلة نعيد نفس الخطوات بس نكت التعليمات :

```
ALTER TABLESPACE TEST
RENAME
DATAFILE 'c:\TEST\yy.DBF'
To 'c:\my\yy.dbf';
```

ثانياً:ALTER DATABASE

طريقة تنفيذ إعادة تسمية او نقل DATAFILE:

١- بعمل shutdown للقاعدة البيانات
٢- نقل أو تغير الاسم فيزيائيا علي مستوى نظام التشغيل
٣- تشغيل DATABASE في وضع MOUNT
٤- تنفيذ التعليمة التالية على مستوى قاعدة البيانات كما يلي :

```
ALTER DATABASE RENAME

FILE  'c:\my\TEST1.DBF'

TO' c:\test\TEST1.DBF';
```

٥- تشغيل DATABASE في وضع open

٩- حذف الTablespace:

الطريقة الأولى مسح الTablespace فقط دون الData Files.

```
DROP TABLESPACE TEST;
```

بهذه الطريقة يظل الData File موجود على نظام التشغيل ويمكن استخدامه عند إنشاء Tablespace جديد .

```
CREATE TABLESPACE OBAY DATAFILE 'c:\TEST\tt.DBF' REUSE;
```

الطريقة الثانية حذف الTablespace ومعه الData File.

```
DROP TABLESPACE TEST
INCLUDING CONTENTS AND DATAFILES;
```

١٠- إنشاء Temporary Tablespace:

كما تحدثنا سابقا إنها مساحة تخزينية تستخدم للتخزين الأغراض المؤقتة وتستخدم في عملية ترتيب البيانات عند امتلاء الجزء المخصص في الذاكرة SORT Area .

```
CREATE TEMPORARY TABLESPACE TEMPEXAM
TEMPFILE 'c:\TEST\TEMPEXAM.DBF' SIZE 7M;
```

بصيغة أخرى :

```
CREATE TEMPORARY TABLESPACE TEMPEXAM TEMPFILE
'D:\TEST\TEMPEXAM.DBF' SIZE 7M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 10M;
```

Default Temporary Tablespace:

- **عند إنشاء مستخدم جديد و عدم تحديد المساحة التخزينية للأغراض الدائمة و المؤقتة فان إي كائن سوف ينشا و إي عملية ترتيب للبيانات سوف تخزن في المساحة الرئيسية System Tablespace وللحل هذه المشكلة بدون إجبار تحديد المساحة Temporary Tablespace عند إنشاء كل مستخدم جديد هو بعمل Default Temporary Tablespace لكل مستخدمي قاعدة البيانات وتمنع استخدام System Tablespace في هذه الحالة ويمكن إنشاء Temporary Tablespace Defaultبطريقتين :**
- **عند إنشاء قاعدة البيانات**
- **باستخدام التعليمة Alter database مثال :**

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE TEMPEXAM;
```

- **ملاحظة:**
- **هذه Default Temporary Tablespace لا يمكن حذفها الا إذا عينت واحدة اخرى مكانها**
- **لا يمكن تحويلها OFFLINE**
- **و لا يمكن تحويلها الي مساحة تخزينية للأغراض الدائمة PERMANENT TABLESPACE**

١١- إنشاء UNDO Tablespace:

تستخدم هذه المساحة للتخزين كل العمليات التي يمكن التراجع عنها باستخدام التعليمة Rollback.

الصيغة العامة :

```
CREATE UNDO TABLESPACE tablespace_name
[DATAFILE C1]
EXTENT MANAGEMENT LOCAL
[AUTOALLOCATE | UNIFORM [SIZE number [K|M] ] ]
```

مثال :

```
CREATE UNDO TABLESPACE undo1
Datafile  'c:\TEST\dataundo.DBF' SIZE 100m;
```

هل يستطيع المستخدم أن يستخدم أكثر من فضاء؟

نعم ، ولعمل ذلك اتبع الخطوات التالية :

١- إنشاء الفضاء ألجدولي الأول وليكن باسم Mytablespace1

```
CREATE TABLESPACE   Mytablespace1

DATAFILE 'C:\TEST\data1.DBF' SIZE 100M;
```

٢- إنشاء الفضاء ألجدولي الثاني وليكن باسم Mytablespace2

```
CREATE TABLESPACE   Mytablespace2

DATAFILE 'C:\TEST\data2.DBF' SIZE 100M;
```

تخصيصها للمستخدم اسمه (jaz) باستخدام ALTER USER اذا كان المستخدم موجود من قبل :

```
alter user jaz

default tablespace Mytablespace1

quota 5m on Mytablespace1

quota 5m on Mytablespace2;
```

٣- الدخول للحساب المستخدم السابق وإنشاء الجدول التالي:

```
CREATE TABLE test (no number);
```

أين سيتم تخزين البيانات ؟؟

سيتم تخزين البيانات في الفضاء ألجدولي الافتراضي Mytablespace1 إلا إذا حددنا فضاء آخر في تعليمة الإنشاء .. أي :

```
CREATE TABLE test2(no number)
Tablespace Mytablespace2 ;
```

:QUERYING INFORMATION

للحصول على معلومات حول TABLESPACE يمكن استخدام :

١- DBA_TABLESPACES : يوفر هذا جميع المعلومات عن TABLESPACE في قاعدة البيانات من حيث الاسم والنوعية ونوعية إدارة EXTENT .

```
select * from DBA_TABLESPACES;
```

٢- V$TABLESPACE: يوفر معلومات عن اسم ورقم TABLESPACE

```
select * from  V$TABLESPACE;
```

للحصول على معلومات حول DATAFILE يمكن استخدام:

١- DBA_DATA_FILES: يوفر معلومات عن اسم DATAFILE وال TABLESPACE التي ينتمي إليها و حجم DATAFILE وغيرها من المعلومات .

```
select * from       DBA_DATA_FILES;
```

٢- V$DATAFILE : معلومات حول DATAFILE مثل حالة DATAFILE (online, offline) وحجم DATAFILE وغيرها من المعلومات

```
3-  select * from       V$DATAFILE;
```

للحصول على معلومات حول TEMP FILE يمكن استخدام:

1- DBA_TEMP_FILES: يوفر معلومات عن اسم TEMP FILE وال TABLESPACE التي ينتمي إليها و حجم TEMP FILE وغيرها من المعلومات .

```
select * from  DBA_TEMP_FILES;
```

٢ V$TEMPFILE: معلومات حول TEMP FILE مثل حالة TEMP FILE (online, offline) وحجم TEMP FILE وغيرها من المعلومات

```
select * from  V$TEMPFILE;
```

:Control File

او يمكن أن نسميه ملف التحكم وهو ملف ثنائي أي لا يمكن قراءته إذ أنه ليس نصياً ، وبدونه لا تعمل قاعدة البيانات ، ويمكن قراءته في الوضع Mount ، إذاً يمكن أن نفتح قاعدة البيانات في الوضع Nomount حتى لو فقدنا ملف الControl File .

وبالطبع يكون تابع لقاعدة بيانات واحدة ، ويتم تحديثه فقط عن طريق الOracle Server اثناء عمل قاعدة البيانات فلا يستطيع مدير قاعدة البيانات تحديثه .

إذا فقدنا الControl File نحتاج لاسترجاعه قبل فتح قاعدة البيانات من جديد .

يتم إنشاؤه أثناء إنشاء قاعدة البيانات ، كما يمكن زيادة عدد الControl File بعد إنشاء قاعدة البيانات حتى نستطيع الاسترجاع إذا فقدنا احد الControl File .

ويحتوى الControl Files على الاتي :-

١ـ اسم قاعدة البيانات.

٢ـ وقت إنشاء قاعدة البيانات.

٣ـ اسماء الTablespaces.

٤ـ اسم ومكان الData Files والRedo Log Files.

٥ـ الRedo Log الحلى وكذلك رقمه المتسلسل .

٦ـ ومعلومات عن النسخ الاحتياطي . مثلاً معلومات الRMAN .

٧ـ ومعلومات عن الأرشيف .

Multiplexing the Control File :

والمعنى أن قاعدة البيانات تعمل على عدد من الControl File مع العلم أن كل ملفات ال Control File هي طبق الاصل ، نستطيع بهذه الطريقة تأمين قاعدة البينات فى حال فقدان احد الControl File.

والافضل كذلك توزيع هذه الملفات على عدد من الDisk حتى لا نفقدها جميعاً .

الان نفترض أن لدينا قاعدة البيانات اسمها ORCL تحتوى على ٣ ملفات من الControl Files نريد أن نضيف لها ملف Control File اخر .

سنقوم بعمل هذا السناريو فى حالين :-

الحال الاول : باستخدام الSPFILE.

الحال الثاني : باستخدام الinit.ora.

الحالة الاولى (SPFILE) :-

١- نقوم بتعديل المتغير Control_files على ملف المتغيرات (SPFILE) ونضيف اليه مكان الControl File الذى نريد أن نضيفه لقاعدة البيانات .

```
ALTER SYSTEM SET
CONTROL_FILES='C:\oracle\product\10.1.0\oradata\orcl\CONTROL01.CTL','C:\oracle\product\10.1.0\oradata\orcl\CONTROL02.CTL','C:\oracle\product\10.1.0\oradata\orcl\CONTROL03.CTL','C:\oracle\product\10.1.0\oradata\orcl\CONTROL04.CTL'  SCOPE=SPFILE;
```

2- يلزمنا إغلاق قاعدة البيانات الان حتى يتم تحديث التعديل السابق .

```
Shutdown immediate;
```

٣- عن طريق نظام التشغيل نقوم بعمل نسخ لاحد ملفات الControl Files ال موجودة الان ونضعه ونسميه كما حددناه فى المتغير Control_Files

٤- الان يمكن فتح قاعدة البيانات .

```
STARTUP;
```

وهكذا قاعدة البيانات الان تعمل بأربعة من الControl Files بدلاً من ثلاثة .

ومن أجل التأكد من إن قاعدة البيانات الان تعمل بأربعة من الControl Files بدلاً من ثلاثة نكتب التعليمة التالية لعرض جميع ملفات Control Files :

```
Show parameters Control Files;
```

لاحظ أن العملية تتطلب إغلاق لقاعدة البيانات ومن ثم عمل نسخ على نظام التشغيل ومن ثم فتح قاعد البيانات من جديد .

الحالة الثانية (init.ora) pfile :-

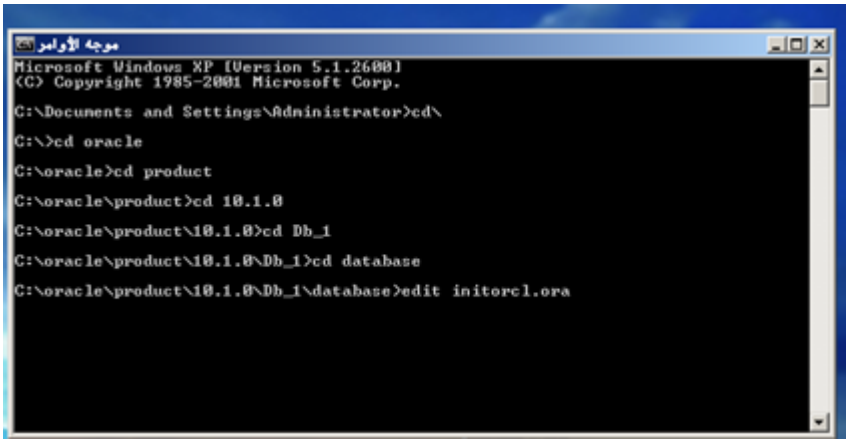
في الحالة الاولى قاعدة البيانات كانت تعمل على ملف المتغيرات (SPFILE) والان نفترض أن قاعدة البيانات تعمل على ملف المتغيرات (initorcl.ora) :-

١- قفل قاعدة البيانات .

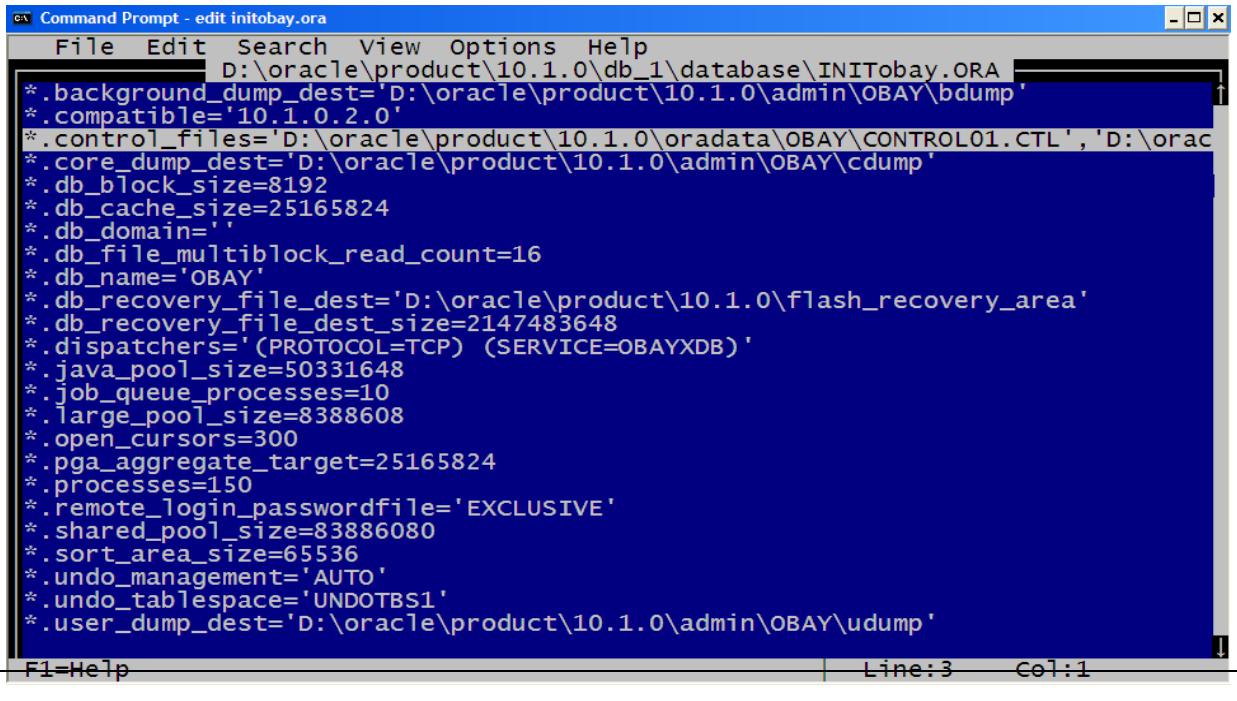
```
Shutdown immediate;
```

٢- تعديل المتغير Control_Files على ملف المتغيرات initorcl.ora.

وذلك عن طريق نظام التشغيل .



عند كتابة الامر اعلاه سيفتح ملف المتغيرات ونعدل فى المتغير Control_Files بحيث نكتب المسار الجديد واسم الملف الجديد ، ولنفترض أن اسمه CONTROL05 فى نفس المسار .



بعد التعديل نقوم بحفظ التعديلات على الملف .

3- عن طريق نظام التشغيل نقوم بعمل نسخ لاحد ملفات الControl Files الموجودة الان ونضعه ونسميه كما حددناه فى المتغير Control_Files .

```
STARTUP;
```

بالطبع يمكن التحقق من التعديل عن طريق الامر SHOW APRAMAETERS.

إنشاء ملف تحكم جديد Create New Control Files

متى احتاج Control Files :

- عند خراب Control Files
- عند انشاء قاعدة البيانات وعملنا ان عدد الملفات DataFile 50 بعدها اردنا عمل DataFile 50 في قاعدة البيانات ولعمل ذلك لابد من Control Files جديد(علل:وظيفة)
- عندما نريد تغيير تغيير اسم قاعدة البيانات .

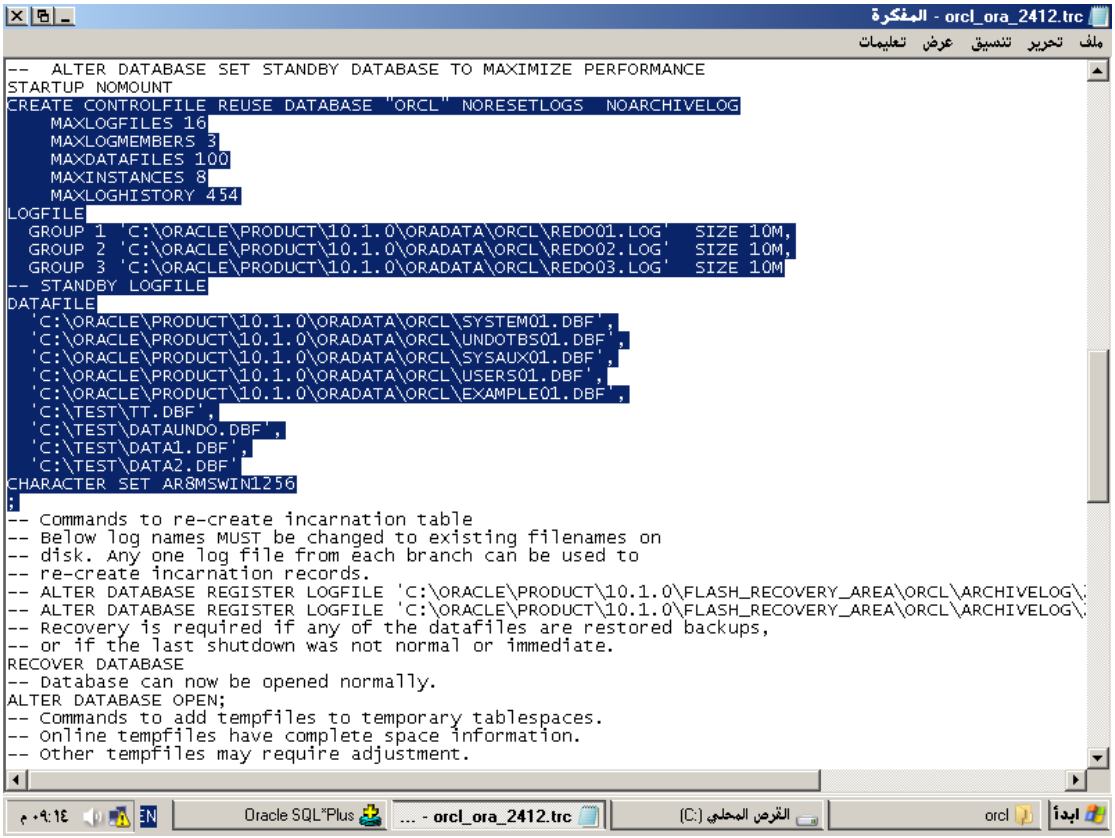
خطوات إنشاء Control Files جديد:

لأنشاء Control Files يجب ان تكون عارف قيم المتغيرات و اسماء الملفات و مواقعها الموجودة في قاعدة البيانات ولذلك يمكننا توليد تعليمة انشاء Control Files من قيل اوراكل باستخدام التعليمة :

```
Alter DataBase backup ControlFile to Trac;
```

تقوم بتوليد تعليمة انشاء Control Files في ملف نصي الموجود

```
C:\oracle\product\10.1.0\admin\orcl\udump
```



١- نقوم بإغلاق قاعدة البيانات :

```
Shutdown Immediate;
```

2- تشغيل قاعدة البيانات بوضع (NOMOUNT):

```
STARTUP NOMOUNT;
```

3- انسخ الجزء المظلل و لصقه :

```
CREATE CONTROLFILE REUSE DATABASE "ORCL" NORESETLOGS NOARCHIVELOG
  MAXLOGFILES 16
  MAXLOGMEMBERS 3
  MAXDATAFILES 100
  MAXINSTANCES 8
  MAXLOGHISTORY 454
LOGFILE
```

```
 GROUP 1 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO01.LOG'
SIZE 10M,

 GROUP 2 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO02.LOG'
SIZE 10M,

 GROUP 3 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO03.LOG'
SIZE 10M

-- STANDBY LOGFILE

DATAFILE

 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF',

 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF',

 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF',

 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF',

 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF',

 'C:\TEST\TT.DBF',

 'C:\TEST\DATAUNDO.DBF',

 'C:\TEST\DATA1.DBF',

 'C:\TEST\DATA2.DBF'

CHARACTER SET AR8MSWIN1256;
```

4-نقوم بتشغيل قاعدة البيانات باستخدام التعليمة

```
Alter database open;
```

:QUERYING INFORMATION

للحصول على معلومات حول Control Files يمكن استخدام :

١- V$ControlFILE : يرجع اسم وحالة كل controlfile

```
select * from V$ControlFILE;
```

٢- V$PARAMETER : يرجع اسماء parameters التي تشتغل عليهن قاعدة البيانات :

```
select * from V$PARAMETER;
```

٣- V$Controlfile_Record_Section : يرجع معلومات عن control file المستخدمة في الجلسة.

```
select * from V$Controlfile_Record_Section;
```

٤- Control_files:يرجع اسماء واماكن ملفات التحكم

```
show parameters control_files;
```

:Redo Log Files

يستخدم هذا الملف لتسجيل التغييرات التي تحصل على البيانات بحيث يتم تسجيل التغييرات التي تم تثبيتها او لا ، نستفيد من هذا الملفات في استرجاع قاعدة البيانات في حال حدوث فشل . بحيث تكون هناك بعض البيانات لم تكتب في الData Files بينما كتبت في هذا الملفات .

:Redo Log Buffer

وهو الجزء الموجود في الذاكرة في الجزء SGA وهي المعلومات التي تكتتب بواسطة العملية LGWR LGWR Background Process يقوم بكتابة البيانات الموجود الRedo log Buffer الى ال Redo Log Files.

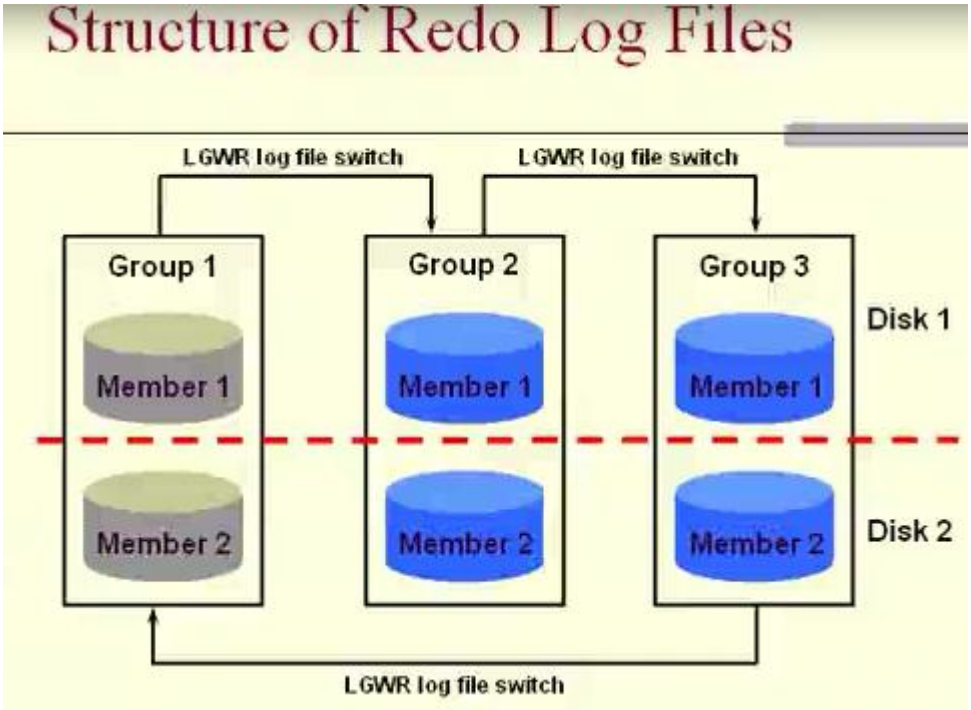
Redo Log Files مقسمة الي مجموعات كل مجموعة يمكن ان يكون فيها اكثر من ملف لكن المجموعة تعبر عن ملف واحد لان الملفات التي فيها عبارة عن نسخ من نفس الملف

،بحيث تعمل كل قاعدة البيانات على الاقل باثنين من الGroups، وكل ملف فيها يسمى member إي عضو من هذه المجموعة

ولضمان الحفاظ على هذا الملف فإن كل Group ينتظم في شكل Members وكل ال Member داخل الGroup هي نسخة طبق الاصل الهدف منها تقليل نسبة خطر الفقدان .

كما ان Redo Log Files تحتوي على كل عمليات المناقلات transaction التي عملها commitو التي لم يعمل لها commit لأنه مثل المسجل يسجل كل عمليات التي تجري على البيانات ،لكن ان عند عمل commit للمناقلات فيه عندنا عداد اسمة System change number وهو عبارة عن رقم يزيد و يقوم بإضافة رقم مخصص في ملف redo عند عمل commit لعملية المناقلة، إي أن اوراكل قادر يعرف إي transaction عملة commit ومن لم يعمله commit.

Structure of redo log file بنية



نلاحظ ان كل مجموعة تتكون من ملفين في حال انضرب واحد نستخدم الثاني لكن المشكلة عند انضراب الاثنين فان اوراكل يعلق .

حيث عند الكتابة على اوراكل يتم الكتابة في member1 و member٢ من المجموعة ١ وعند امتلاء يتم الانتقال الي member1و member٢ من المجموعة ٢ فان اوراكل يقوم بعمل log switch ويزيد log sequins بمقدار واحد عند كل انتقال بين المجموعات

ملاحظة: (log sequins : هو عدد الانتقالات من كل مجموعة الي اخرى).

و عند حدوث log file switch يتم تنفيذ عملية checkpoint مع كل عملية log switch وكذلك نقل معلومات عنها في control file .

LGWR Background Process

يقوم بكتابة البيانات الموجود الRedo log Buffer الى الRedo Log Files فى الحالات التالية :-

١- لحظة تثبيت العمليات (Commit).

٢- لحظة امتلاء الBuffer الى الثلث .

٣- عندما يكون هناك اكثر من ميجا بايت من السجلات في buffer تنقلها قبل عملية DBWn.

كما يقوم LGWR بكتابة البيانات

كما يقوم بكتابة البيانات الموجود الRedo log Buffer الى الRedo Log Files

لان عند عمل Log switch يحدث عملية Check point التي تعني نقل كل ما هو في الذاكرة الي قاعدة البيانات و كذلك تحدث عند اغلاق قاعدة البيانات بشكل طبيعي او عند عمل Check point يدوياً

و نستطيع الاحتفاظ ببيانات المجموعات طالما لم يتم الانتقال من المجموعة الاخيرة الي الاولى لانها تقوم بحذف البيانات السابقة وكتابة الجديدة.

و يمكننا تحديد عدد المجموعات و عدد اعضاء كل مجموعة في قاعدة البيانات عند انشائها وذلك بتحديد ال parameter التالية:

Maxlogfiles 32

هذا يعنى انه يمكننا انشاء ٣٢ مجموعة في قاعدة البيانات

Maxlogmembers 2

هذا يعني ان كل مجموعة يمكن ان تحتوى على ملفين فقط

إذاً عملية الLog Switch تحدث الياً دون تدخل من مدير قاعدة البيانات ، لكن قد يحتاج مدير قاعدة البيانات احياناً لعمل هذه المهمة يدوياً من خلال الامر التالي :-

```
ALTER SYSTEM SWITCH LOGFILE;
```

كذلك عملية الCheckpoint تحدث الياً عند حدوث الLog Switch وهى تقوم بعمل تعديل للHeader لكل الData Files والControl Files للإشارة بأن اكمل بنجاح .

ايضاً يمكن عمل Checkpoint الياً عن طريق مدير قاعدة البيانات بالأمر التالي :-

```
ALTER SYSTEM CHECKPOINT;
```

إضافة Online Redo Log Files Group :

فى بعض الاحيان نحتاج لإضافة Online Redo Log Files Group جديد ، وذلك عن طريق الامر التالى :-

```
ALTER DATABASE ADD LOGFILE GROUP 5

'c:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\REDO05.RDO' SIZE 10M;
```

ولإضافة Members للGroups :-

```
alter database add logfile member

 'c:\oracle\product\10.1.0\oradata\ ORCL \red025.rdo'  to  group  5,

'c:\oracle\product\10.1.0\oradata\ ORCL \red06.rdo'  to  group  5;
```

حذف Online Redo Log Groups:-

```
ALTER DATABASE DROP LOGFILE GROUP 5;
```

مع ملاحظة أن قاعدة البيانات تعمل على الاقل على اثنين من الGroups ، كذلك لا يمكن حذف current or active Groups .

Current: يعني اوراكل يكتب عليها

Active: يعني بياناتها لم تنتقل للقاعدة البيانات

أيضاً عند حذف الGroup عن طريق الOracle تظل الملفات موجودة على مستوى نظام التشغيل .

حذف Online Redo Log Members

```
ALTER DATABASE DROP LOGFILE MEMBER

''C:\ORACLE\PRODUCT\10.1.0\ORADATA\orcl\RED025.RDO';
```

Clearing مسح البيانات:

تعني مسح جميع البيانات التي تحتويها ونستفيد منها في

حل مشكلة العطل الذى يحدث لكل الMembers فى الGroup ، فهي ضمناً تعنى إعادة إنشاء لل Redo Log Files .

```
ALTER DATABASE CLEAR LOGFILE GROUP 2;
```

إذا كانت قاعدة البيانات تعمل فى الوضع Archive log فإن عملية الClearing اعلاه ستفشل إذا كان الGroup 2 لم تتم ارشفته ، عموماً سنتحدث عن موضوع الارشيف لاحقاً ولكن لحل

هذا الإشكال يمكن تعديل الامر السابق للاتي :-

```
ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP 5;
```

إعادة تسمية الRedo Log Files:-

يمكن إعادة تسمية الRedo Log Files او تغيير مكانه عن طريق إضافة Log File جديد وحذف القديم . يمكن كذلك استخدام الامر ALTER DATABASE RENAME FILE لكن هذا الامر يتطلب جلب قاعدة البيانات فى الوضع MOUNT.

١- UNDO MANAGEMENT :

في المحاضرة السابقة تحدثت عرضاً عن الUndo Tablespace وذكرت أنه يستخدم من قبل الOracle Server لتخزين الUndo Information ، لكن هل فكرت يوماً فى عملية الROLLBACK ؟ وهل سألت نفسك كيف يمكنك التراجع عن عمليات التعديل التي قمت بها فى بعض الحقول؟ ، رغم أنه تم تغير القيم القديمة الى القيم الجديدة .

السناريو باختصار أنه عند إجراء عملية تعديلات على البيانات فى قاعدة البيانات فإن ال Oracle Server يقوم يتخزين القيم القديمة فى الUndo Tablespace قبل ان يتم تغييرها بالقيم الجديدة ، هذا السناريو يتيح لنا فرصة التراجع عن العمليات متى احتجنا إلى ذلك بشرط أن تكون القيم القديمة ما زالت موجودة فى الUndo Tablespace إذ لا يتصور أن تظل هذه القيم موجودة إلى ما لانهاية فهو فى اخر الامر Tablespace له سعة محدودة وكذلك له فترة احتفاظ بالمعلومات محدودة يقوم بتهيئتها مدير قاعدة البيانات حسب ما يراه مناسباً ، نستطيع كذلك من خلال هذا السناريو ان ننجز عملية الFlashback Queries إذا توفرت القيم القديمة سنتحدث عن الFlashback Queries في هذه المحاضرة ولكن عموماً هو عبارة عن استعلام عن قيم لكن فى الماضي عند زمن معين .

فالUndo Tablespace تتم إدارته عن طريق Locally Managed Tablespace وايضاً Automatic Extent allocation ، فلحظة بدء العملية يتم تخصيص وإنشاء Undo Segment الياً لتخزين القيم التى تتغير فى قاعدة البيانات بواسطة هذه العملية ، بحيث يتم تخصيص Undo Segment واحدة لكل عمليات ولكن يمكن أن تخدم هذه الSegment عدد من العمليات، فعند امتلاء الExtent يتم التحول للتى بعدها فى نفس الSegment بحيث يكون على الاقل هناك اثنين من الExtents فى كل Segement أما الحد الاقصى فيعتمد على DB Block Size ، أما إذا تم ملء جميع الExtents فى الUndo Segment فإنه يتم إعادة الكتابة فى الExtent ابتداء من الأول ، أو يتم طلب تخصيص Extent جديدة.

قد يكون فى قاعدة البيانات الواحدة اكثر من Undo Tablespace لكن لا يمكن ان يعمل فى اللحظة الواحدة اكثر من Undo Tablespace ويتم تحديد عمل الUndo Tablespace فى قاعدة البيانات بواسطة المتغير Undo_Tablespace.

```
SHOW PARAMETER UNDO_TABLESPACE;
```

لإنشاء Undo Tablespace جديد

```
CREATE UNDO TABLESPACE ORCLUNDO DATAFILE
'C:\oracle\product\10.1.0\oradata\ORCL\ORCLUNDO.DBF' SIZE 20M EXTENT
MANAGEMENT LOCAL AUTOALLOCATE;
```

الان قاعدة البيانات تحتوى على اثنين من الUndo Tablespace ولكن يعمل واحد فقط هو UNDOTBS1 كما موضح اعلاه .

ولكن يمكن تغيير الUndo Tablespace الذى يعمل فى قاعدة البيانات من UNDOTBS1 الى ORCLUNDO.

```
ALTER SYSTEM SET UNDO_TABLESPACE='ORCLUNDO' SCOPE=BOTH;
```

قاعدة البيانات اواركل Oracle Database10g تتيح خيار ادرة الياً للUndo Tablespace وهو Automatic Undo Management ويتم تهيئته بواسطة المتغير UNDO_MANAGEMENT بحيث يأخذ القيمة AUTO ، أما خيار الإدارة اليدوي Manual Undo Magamenet فهو خيار مكلف ويحتاج لعمل اكثر من مدير قاعدة البيانات .

أما خيار الإدارة Automatic Undo Management فهو يقلل العبء من مدير قاعدة البيانات بحيث تكون إدارته فقط على مستوى الTablespace.

مدير قاعدة البيانات ينتظره تهيئة المساحة المناسبة للUndo Tablespace وذلك حسب المعلومات التى سيتم تخزينها ، كذلك لابد من تهيئة فترة الاحتفاظ Undo Retention للبيانات فى الUndo Segment وذلك بواسطة المتغير UNDO_RETENTION الذى يأخذ قيمة هي فترة الاحتفاظ بالثواني .

الاصل في هذا المتغير هو أن يأخذ القيمة ٠ وهى تعنى Automatic اى يعنى محاولة الاحتفاظ بالمعلومات حتى تنتهى ، على أن يتم الاحتفاظ بالمعلومات على الاقل ١٥ دقيقة ، لكن يمكن تحديد

قيمة اخرى هى فترة الاحتفاظ بالثواني واقصى قيمة هي 2^{32}.

```
SHOW PARAMETER UNDO_RETENTION;
```

وعموماً Undo information اى المعلومات الموجودة فى الUndo Segments تنقسم الى ثلاث حالات :-

١- Uncommitted Undo Information: وهى المعلومات التى لم يتم تثبيتها إلى الان وذلك لأن العمليات مازالت مستمرة ، هذا النوع من المعلومات لا يمكن حذفها واعادة الكتابة فيها .

٢- Committed Undo Information: وهى لا نحتاجها لعمليات مستمرة ، ولكن لأن فترة الاحتفاظ لم تنتهى بعد "Unexpired"، هذا النوع من العمليات نحتفظ به قدر الإمكان ما لم يؤدى ذلك لفشل بعض العمليات نتيجة عدم وجود مساحة فى الUndo Tablespace فى هذه الحالة يتم اعادة الكتابة فى هذه المعلومات ، ولكن قد يقوم مدير قاعدة البيانات بتهيئة الUndo Tablespace بحيث نضمن عدم مسح واعادة الكتابة فيها وذلك باستخدام الخيار Guaranteeing Undo Retention.

```
ALTER TABLESPACE ORCLUNDO RETENTION GUARANTEE;
```

هذا الخيار بالطبع غير متاح سواء للUndo Tablespace.

هكذا نضمن عدم مسح المعلومات التي لم تنتهى فترة احتفاظها حتى لو أدى ذلك لفشل بعض العمليات لعدم وجود مساحة في الUndo Tablespace.

٣- Expired Undo Information: وهى لا نحتاجها لعمليات مستمرة ، وكذلك فترة الاحتفاظ بها انتهت فيمكن اعادة الكتابة فيها متى ما احتجنا لمساحة فى الUndo tablespace.

من المشاكل التى تحدث كثيراً و التي يجب لمدير قاعدة البيانات مراعاتها :-

١- مشكلة المساحة Undo Tablespace Space Error: ويجب على مدير قاعدة البيانات مراقبة مساحة الUndo Tablespace فالعمليات التى لا تجد مساحة فى الUndo Tablespace تعطى رسالة الخطأ.

ORA-01650: unable to extend rollback segment

٢- Snapshot too old” Error“: وهذا الخطأ يظهر عند تنفيذ استعلام يحتاج لمعلومات Undo Information قد تم مسحها واعادة الكتابة فيها ، لذا يجب على مدير قاعدة البيانات مراعاة فترة الاحتفاظ المناسبة UNDO_RETENTION وكذلك المساحة المناسبة مع مراعاة Guaranteeing Undo Retention.

A- Flashback Query:

هذا النوع نستطيع من خلاله عمل استعلام فى قاعدة البيانات ليس للبيانات الموجودة حالياً وإنما لبيانات موجودة فى لحظة زمنية في الماضي ، ويعتمد هذا النوع اساساً على الUndo Tablespace ، فبمجرد عمل الاستعلام يتم البحث عن المعلومات فى ال Undo Segments ، هذه المعلومات التي ستعرض للمستخدم تظل مؤقتة ومتاحة فقط للSession الحالية ، بالطبع فقد تفشل عملية الFlashback Query إذا لم نجد البيانات المطلوبة فى الUndo Tablespace مثلاً بسبب طول المدة وإعادة كتابة بيانات جديدة فى البيانات المطلوبة لذا على مدير قاعدة البيانات تحديد فترة احتفاظ Retention مناسبة.

ولنتصور الان السناريو التالى حتى نستوعب عملية الFlashback Query بصورة اوضح:

لدينا الان جدول يسمى student يحتوى على مجموعة من الحقول.

```
Create table student (no number,name varchar2(100));

Insert into student values (1,'ail');

Insert into student values (2,'omar');

Insert into student values (3,'mona');

SELECT * FROM student;
```

الجدول student يحتوى الان على ٣ طلاب.

لنفترض أن المستخدم حذف طالب .

```
DELETE student WHERE NO =2;
```

```
Commit;
```

الان لو قمنا باستعراض الجدول.

بقى لدينا فقط ٢ طلاب .

ماذا لو قمنا الان بعمل استعلام على الجدول ولكن فى فترة زمنية فى الماضي ولنفترض انها قبل ٢٠ دقائق .

بالطبع سيبحث عن هذه المعلومات في الUndo segements ويقوم بعرضها ، أما إذا لم يجدها فستظهر رسالة خطأ.

اى هذه المعلومات غير متوفرة حالياً في الUndo Tablespace.

```
SELECT * FROM student AS OF TIMESTAMP TO_TIMESTAMP ('26-01-11:22-20-00','DD-MM-YY:HH24-MI-SS');
```

نتيجة الاستعلام ، عرض ٣ طلاب ، لكن ما هو متاح حالياً هو ٢ طلاب فقط.

هكذا قمنا بعرض معلومات فى الماضي ، لكن ماذا لو أردنا أن نأتي بهذه المعلومات فى الحاضر أي تظل هذه المعلومات موجودة فى الجدول ، يمكن أن ننشئ جدول جديد من المعلومات المستعرض ومن ثم نقوم بحذف الجدول student وإعادة تسمية الجدول الجديد الى student1.

```
CREATE TABLE student1 AS (SELECT * FROM student AS of TIMESTAMP TO_TIMESTAMP ('26-01-11:22-20-00','DD-MM-YY:HH24-MI-SS'));
```

الان نقوم بحذف الجدول student.

```
DROP TABLE student;
```

ثم نقوم بإعادة تسمية للجدول student1 إلىstudent.

```
CREATE TABLE student  AS (SELECT * FROM student1  AS of TIMESTAMP TO_TIMESTAMP ('26-01-11:22-20-00','DD-MM-YY:HH24-MI-SS'));
```

الان الجدول student يحتوى على المعلومات التى كانت به منذ ٢٠ دقيقة.

B-رجوع الجداول الي الزمن الماضي Flashback Table:

ويسمى ايضاً Flashback Table Query واستحدث هذا النوع فى الإصدار Oracle10g ويعتمد اساساً على الUndo Tablespace ، وهو الرجوع بالجدول الى فترة زمنية فى الماضي ويعتمد اساساً الى الذهاب الى الUndo Segments وارجاع الجدول الى لحظة معينة فى الماضي مستفيدا من المعلومات الموجودة Undo Segments ، فقد تكون جرت على الجدول عدد من التعديلات سنجدها فى الUndo Tablespace ، بالطبع قد تفشل عملية الFlashback Table إذا لم نجد المعلومات المطلوبة فى الUndo Tablespace نتيجة لانتهاء فترة الاحتفاظ بالمعلومات المطلوبة مثلاً.

عملية الFlashback Table ليست مضمونة النجاح ، فقد تفشل في احد الأسباب التالية :-

١- إذا تم انتهاك احد القيود Constraint Violated.

٢- إذا لم يتم عمل Enable Row Movement للجدول المطلوب استرجاعه.

٣- إذا لم تتوفر المعلومات المطلوبة فى الUndo Tablespace ، :ORA-08180

no snapshot found based on specified time"

٤- لا يمكن عمل Flashback Table لجدول فى Sys Schema.

آلية التعامل معها :

لنفترض ان لدينا جدول الطلاب Student في المستخدم flashtest فية بيانات خمسة طلاب

```
SELECT * FROM Student;
```

لنضف طالب جديد الى جدول الطلاب

```
INSERT INTO Student VALUES(1,'salah');

INSERT INTO Student VALUES(2,'ali');

Coomit;
```

نقوم باستعراض التاريخ الان:

```
SELECT SYSDATE FROM DUAL;
```

نحتاجة لعمل الFlashback Table.

الان نقوم بحذف الطالب الجديد :

```
DELETE from student WHERE no=2;

Commit;
```

الان نقوم بأخبار اوراكل عن إمكانية التراجع عن التعديلات التي حصلت على هذا الجدول من خلال الامر:

```
ALTER TABLE student ENABLE ROW MOVEMENT;
```

الان لنقوم بعمل Flashback Table للجدول studentإلى ما قبل حذف الحقل NO=6 ،

```
flashback table student to timestamp to_timestamp('02-02-11:20-33-10','dd-mm-
yy:hh24-mi-ss');
```

للتأكد نقوم باستعراض بيانات الطلاب في student:

```
SELECT * FROM Student;
```

C- Flashback Versions Query:

قد تحتاج احياناً كمدير لقاعدة بيانات لمراقبة بعض التعديلات على الحقول فتحتاج لمعرفة القيم التى اخذها حقل معين لفترات تاريخية معينة ، فمثلا قد تتحول مرتبات الموظفين كل فترة فيتم تعديل ملف الSalary كل فترة زمنية ، ولكن لنفترض أنى احتاج لمعرفة الHistory او القيم السابقة للمرتب لموظف معين منذ أن تم تعينه ، فكل ما هو متاح لى فى الجدول الان هو اخر قيمة للمرتب الحالي.

وقد استحدث هذا النوع فى الإصدار Oracle10g ويعتمد اساساً على الUndo Tablespace. بحيث يأتي بالمعلومات التي يريدها من الUndo Segments بعد أن نحدد له الفترة الزمنية المحددة أو عن طريق System Change Number (SCN).

```
create user fbuser identified by fbuser;

grant connect,resource,dba to fbuser;

conn fbuser/fbuser;

        create table flashtest(id number,name varchar2(20),sal number);

        insert into flashtest(1,'ali',100);

       select * from flashtest;

       update flashtest set sal=500;

       commit;

       update flashtest set sal=600;

       commit;
```

```
versions_startscn,versions_endscn,versions_xid,versions_operation,id,name,sal  select
from flashtest VERSIONS between SCN MINVALUE and MAXVALUE where id=1;
```

E- Flashback Drop:

فى الإصدارات السابقة من قاعدة البيانات اوراكل Oracle Database لحظة عمل حذف للجدول Drop Table فإنه سوف يزال ايضاً من الData Dictionary وسوف نقوم بعمل استرجاع Recovery لقاعدة البيانات إذا أردت أن استرجع جدول واحداً تم حذفه خطأً ، لا شك أنه حل مكلف ومكلف جداً من حيث الزمن ومن حيث أننا سأفقد بعض العمليات.

فعند حذف جدول فى الإصدارات السابقة من اوراكل يتم تحرير المساحات التي كان يستقلها ، أما في الإصدار Oracle10g لا يتم تحرير المساحات التي يستقلها الجدول وما يتبعه من كائنات لحظة حذف الجدول ، وإنما يتم وضع الجدول مؤقتاً في سلة المهملات RECYCLE BIN ويظل الجدول مملوكاً ايضاً للمستخدم رغم انتقاله الى سلة المهملات ، ولكن بالطبع لحظة انتقاله إلى سلة المهملات يتم إعادة تسميته حتى لا يحدث تضارب فى الاسماء إذا أردت أن اقوم بإنشاء جدول اخر بنفس الاسم ، هذا الوضع يتيح لنا فرصة استرجاع الجدول من سلة المهملات دون الحاجة لاسترجاع قاعدة البيانات ، مما يقلل لنا فترة الاسترجاع وكذلك دون أن نفقد أي عمليات .

بكل بساطة سلة المهملات RECYCLE BIN فى قاعدة البيانات اوراكل تشبه الى حد كبير سلة المهملات فى نظام التشغيل ويندوز WINDOWS.

يمكن الاستعلام عن سلة المهملات RECYCLE BIN بواسطة:

DBA_RECYCLEBIN

USER_RECYCLEBIN

كما يمكن الاستعلام عن طريق الامر:

SHOW RECYCLEBIN

آلية التعامل :

يقوم المستخدم بحذف الجدول student، ومن ثم نقوم باستعلام عن سلة المهملات RECYCLE BIN.

```
DROP TABLE student;
```

استعلام عن الجداول في سلة المحذوفات:

```
SHOW RECYCLEBIN;
```

استرجاع جدول من سلة المحذوفات

لاحظت معي كيف أنه رغم أننا قمنا بحذف الجدول USER_MASTER إلا أن ما زال موجوداً فى سلة المهملات ويمكننا كذلك استرجاعه .

```
FLASHBACK TABLE student TO BEFORE DROP;
```

استعراض جميع الجداول الموجودة:

```
Select * from tab;
```

لقد قمنا باسترجاع الجدول من سلة المحذوفات لكن قد تفشل عملية الاسترجاع إذا قمنا بإنشاء جدول بنفس الاسم قبل عملية الاسترجاع لذلك يمكن اعادة تسميته لتفادى عملية تضارب الاسماء.

```
FLASHBACK TABLE student TO BEFORE DROP RENAME TO studen1;
```

لقد قمنا باسترجاع الجدول ولكن باسم جديد .

```
SELECT * FROM USER_RECYCLEBIN;
```

لا يوجد جدول في سلة المهملات .

يمكن تنظيف سلة المهملات ومسح الجداول من سلة المهملات بواسطة الامر PURGE.

```
PURGE TABLE student1;
```

لا يوجد الان الجدول student1 فى سلة المهملات .

كذلك يمكن حذف الجدول عن طريق المستخدم دون وضع الجدول فى سلة المهملات وانما حذفه مباشرة من قاعدة البيانات بواسطة الامر DROP TABLE <TABLE_NAME> PURGE

```
DROP TABLE student PURGE;
```

إذاً من الحالات التي يتم فيها حذف الجداول من قاعدة البيانات دون المرور بسلة المهملات :-

١- استخدام الامر DROP TABLE <TABLE_NAME> PURGE.

٢- لحظة حذف الTABLESPACE الذى يحوى الجداول بواسطة الامر

DROP TABLESPACE <TABLESPACE_NAME> INCLUDING CONTENTS.

٣- عند حذف المستخدم الذى يحوى الجداول بواسطة الامر

```
DROP USER <USER_NAME> CASCADE.
```

F- Flashback Database:

هل فكرت يوماً ماذا تفعل لو قمت بحذف مستخدمٍ ما خطأً ؟ الجواب وبكل بساطة اقوم بعمل استرجاع لقاعدة البيانات Recovery من النسخ الاحتياطي Backup.

هذا الحل صحيح ولكن مكلف من حيث الزمن خصوصاً إذا كانت قاعدة البيانات كبيرة ، فاسترقاع احجام كبيرة من الملفات لا بد أنه مكلف من حيث الزمن .

ما الحل إذاً ، الحل هو Flashback Database.

الFlashback Database هي إرجاع قاعدة البيانات الى نقطة زمنية معينة فى الماضي ، بحيث جميع التعديلات التي حدثت فى قاعدة البيانات من تلك النقطة الى الان كأن لم تكن.

نستفيد من الFlashback Database فى حالة حدوث اخطاء منطقية ، كأن يحذف مدير قاعدة البيانات مستخدم عن طريق الخطأ أو عمل Truncate Table ، أما الأخطاء الفيزيائية كأن نفقد Data File او غيره من الملفات فلا نستطيع الاستفادة من الFlashback Database وكل ما نستطيع هو عمل استرجاع للملفات من النسخ الاحتياطي بالطريقة التقليدية.

لحظة عمل Enable Flashback Database يتم إنشاء Background Process يسمى Recovery Writer (RVWR) ويتم تخصيص جزء فى الذاكرة SGA يسمى Flashback Buffer ويتم تخصيص جزء من الديسك للFlashback logs وهذا الجزء هو دائماً الFlash Recovery Area .

بحيث يتم وضع كل Block تم تعديله فى قاعدة البيانات الى الFlash Buffer بعد ذلك يقوم الRVWR بوضع هذه الBlocks من الFlashback Buffer إلى الFlashback Logs .

كما ذكرت أنه ليست كل التعديلات تكتب فى الFlashback Buffer وإنما Complete Block Images ، لذا فإن هناك كثير من التعديلات تحدث ولا يتم كتابتها مباشرة فى الFlashback Buffer وإنما يتم تأخيرها وربما تحدثت تعديلات اخرى قبل أن تكتب فى ال Flashback Buffer ، ما أرمى اليه أنا أنه قد يتم كتابة Complete Block وهو يحوى مجموعة مختلفة من التعديلات حدثت فى ازمنة مختلفة ، لذا عملية الFlashback Database لوقت مضبوط هى من الصعوبة بمكان ، لذا فإن الFlashback Database يحتاج لأن تكون قاعدة البيانات فى النمط Archive Log Mode وذلك لأن الارشيف يحتوى على جميع تفاصيل التعديلات التي حدثت فى قاعدة البيانات على خلاف Flashback Logs ، إذا الFlashback Logs مع الArchive Logs يستطيعان أن ينجحا عملية الFlashback Database

لتهيئة الFlashback Database نتبع الخطوات الاتية:-

١- التأكد من أن قاعدة البيانات تعمل فى النمط Archive Log Mode.

تحدثنا فى محاضرات سابقة على أن قاعدة البيانات تعمل على الاقل على اثنين من Redo Log Files وهى ملفات تحوى على التعديلات التى تمت فى قاعدة البيانات .

وأن LGWR Background Process يقوم بكتابة التعديلات الموجودة فى ال Redo Log Buffer ويضعها فى الRedo Log Files ، وتكون عملية الكتابة بشكل دورى بين الRedo Log ويتم إعادة الكتابة فى الRedo Logs ، مما يؤدى لفقدان البيانات الموجودة فيه لذا كان الحل هو تشغيل قاعدة البيانات فى الانمط Archive Log Mode بحيث يتم إنشاء Background Process جديد يسمى ARCn بحيث يقوم

بعمل نسخ لملفات الRedo Log Files بحيث لا نفقد البيانات الموجودة في هذه الملفات و التي نعتمد عليها في عملية الاسترجاع Recovery ؛ والتي بدونها تفشل عملية الRecovery.

لمعرفة هل قاعدة البيانات تعمل في الوضع Archive Log Mode ام لا؟

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

إذا قاعدة البيانات فى الوضع No Archive Log.

لتهيئة قاعدة البيانات فى الوضع Archive Log نقوم بالخطوات الاتية:

A- تحديد المكان الذى سوف نضع فيه الأرشيف بواسطة المتغير LOG_ARCHIVE_DEST مع مراعاة أنه يمكن تحديد ١٠ اتجاهات مختلفة وذلك بواسطة المتغيرات LOG_ARCHIVE_DEST_n حيث n تأخذ القيم من ١ الى ١٠.

```
Archive log list ;

انشاء مجلد في القرص c باسم ARCHIVE

ALTER SYSTEM SET

LOG_ARCHIVE_DEST_1='LOCATION=c:\ARCHIVE' SCOPE=SPFILE;
```

B- نغلق قاعدة البيانات ونفتحها فى الوضع Mount ، ثم ننفذ الامر

ALTER DATABASE ARCHIVELOG.

```
Shutdown immediate;
```

```
startup mount

Archive log list

ALTER DATABASE ARCHIVELOG;
```

الان قاعدة البيانات تعمل فى النمط ARCHIVELOG MODE .

يمكن الاستعلام ايضاً عن عمل ARCHIVELOG MODE بواسطة:

```
SELECT ARCHIVER FROM V$INSTANCE;
```

لحظة عمل SWITCH LOGFILE يتم عمل نسخ لملف الREDO LOG FILE للأرشيف.

```
ALTER SYSTEM SWITCH LOGFILE;
```

كما يمكن الاستعلام عن ملفات الارشيف

```
SELECT NAME FROM V$ARCHIVED_LOG;
```

هذه هي ملفات الأرشيف.

٢- تهيئة الFlash Recovery Area:

وهو عبارة عن مساحة على الديسك يتم التحكم فيها عن طريق الاوراكل بحيث يتم فيها تخزين ملفات النسخ الاحتياطي ويتم إدارة هذه المساحة والتحكم فيها عن طريق الاوراكل ويتم تحديد هذه المساحة بواسطة المتغير DB_RECOVERY_FILE_DEST وهو لتحديد المسار وأيضاً المتغير DB_RECOVERY_FILE_DEST_SIZE لتحديد حجم هذه المساحة، أما عملية إدارة هذه المساحة فتكون عن طريق الاوراكل حيث يتم تحديد فترة الاحتفاظ بملفات النسخ الاحتياطي وأيضاً حذف الملفات القديمة المنتهية وزيادة حجم هذه المساحة إذا تطلب الأمر وغيرها من إجراءات لإدارة هذه المساحة يتم عن طريق الاوراكل.

والFlashback Logs لا تتم إدارته إلى فى ال ، إذاً لتهيئة الFlash Recovery Area يلزمنا تهيئة المتغيران (DB_RECOVERY_FILE_DEST & DB_RECOVERY_FILE_DEST_SIZE)

```
ALTER SYSTEM SET
DB_RECOVERY_FILE_DEST='C:\ORACLE\PRODUCT\10.1.0\FLASH_RECOV
ERY_AREA';
```

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=2G;
```

3- تحديد فترة الاحتفاظ بال Flashback Log:

وذلك قبل أن يتم استخدامه وإعادة الكتابة فيه مرة اخرى ويتم تحديد ذلك بواسطة المتغير DB_FLASHBACK_RETENTION_TARGET وهو زمن الاحتفاظ بالدقائق والاصل أن يتم الاحتفاظ لمدة يوم.

```
ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=1440;
```

```
SHOW PARAMETER DB_RECOVERY_FILE_DEST;
```

4- اغلاق وفتح قاعدة البيانات فى الوضع Mount:

ومن ثم تمكين الFlashback بواسطة الامر ALTER DATABASE FLASHBACK ON

```
ALTER DATABASE FLASHBACK ON;
```

هكذا انتهينا من تهيئة الFlashback.

للتأكد من تهيئة الFlashback نقوم بعمل الاستعلام الاتي:

```
SELECT FLASHBACK_ON FROM V$DATABASE;
```

لمراقبة الFlashback

```
SELECT * FROM V$FLASHBACK_DATABASE_LOG;
```

لمعرفة كيفية استخدام الFlashback Database نتابع معاً:

١- مدير قاعدة البيانات قام بحذف المستخدم TEST عن طريق الخطأ.

```
DROP USER TEST CASCADE;
```

```
select username from dba_users  where  username='test';
```

الان لا يوجد مستخدم اسمه TEST فى قاعدة البيانات.

٢- اغلاق قاعدة البيانات وفتحها فى الوضع Mount ومن ثم عمل Flashback Database للزمن

```
FLASHBACK DATABASE TO TIMESTAMP TO_TIMESTAMP('02-02-11:09-
55-54','DD-MM-YY:HH24-MI-SS');
```

٣- 02_02-11:09-55-54 وهو الوقت ما قبل حذف المستخدم TEST.

٣- نفتح قاعدة البيانات فى الوضع Resetlogs .

```
ALTER DATABASE OPEN RESETLOGS;
```

٤- اخيراً لنتأكد من وجود المستخدم TEST.

```
SELECT USERNAME FROM DBA_USERS WHERE USERNAME='TEST';
```

أما إذا لم نكن نعرف مثلاً وقت حذف المستخدم TEST فليس عندنا حل سواء التخمين ومن ثم فتح قاعدة البيانات فى الوضع Read Only والتأكد من جدوى الاسترجاع وقد نلجأ لإغلاق قاعدة البيانات وفتحها مرة اخرى ومن ثم إعادة الFlashback Database لوقت اخر ، ولان نفتح قاعدة البيانات مباشرة فى الوضع Resetlogs لان الخيار Restlogs يقوم بإعادة تهيئة الLog Sequence Number ويبدأ بأخذ القيمة (٠٠١) من جديد.

قواعد البيانات الموزعة:

١- Oracle Net Services :-

أن يكون لديك مخدم Database Server يحتوى على قاعدة البيانات يتصل به جميع Client Application او جميع الاجهزة التي تحتوى على برامج تحتاج للاتصال بقاعدة البيانات هذا هو الامر الطبيعي والمعتاد والذى تعمل به اغلب الشركات في العالم ، إذ لا يتصور أن تكون جميع الاجهزة التي تحتوى على برامج تحتاج للاتصال بقاعدة البيانات تحتوى ايضاً على قاعدة البيانات ، وإلا فإننا نحتاج لقاعدة البيانات لكل جهاز يحتوى عل برنامج وهذا غير مقبول عقلاً ولا عملاً.

وإذا سلمنا بأن يكون لدينا مخدم يحتوى على قاعدة البيانات Database Server تستطيع جميع الاجهزة التى تحتوى على برامج تحتاج لقاعدة البيانات الاتصال بهذا المخدم ، فإن محور حديثنا فى هذه المحاضرة سينصب على كيفية إنجاح هذا الاتصال .

إذاً الامر سيكون على جانبين :

الجانب الاول : وهو جانب المخدم Database Server وكيف يستطيع خدمة جميع الطلبات التي تصله للاتصال بقاعدة البيانات .

اما الجانب الثاني : وهو جانب الClient Application وهو الجهاز الذى يريد الاتصال بقاعدة البيانات ، وكيف يستطيع الوصول لقاعدة البيانات ؟

والان سنتحدث عن الجانب الاول وهو جانب المخدم Database Server وكيف يستطيع خدمة جميع الطلبات للاتصال بقاعدة البيانات ؟

في هذا الجانب يستطيع المخدم Database Server خدمة طلبات الاتصال بقاعدة البيانات بواسطة

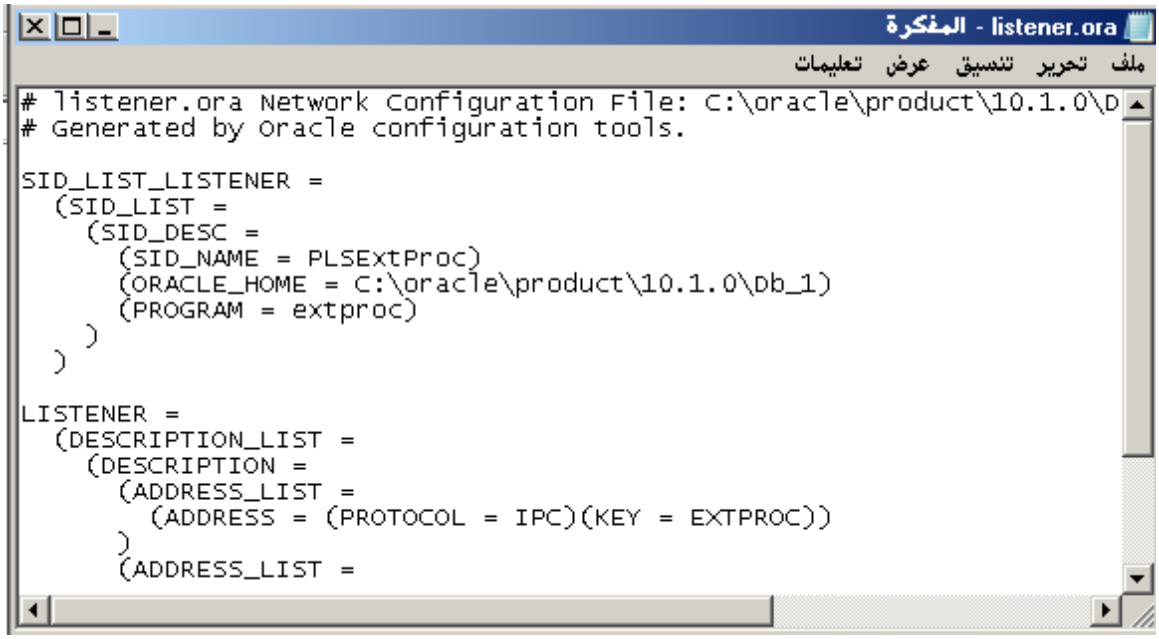
الOracle Net Listener :وهو المسؤول عن عملية معالجة طلبات الClients للاتصال بقاعدة البيانات ، فبدون الListener محاولة عملية الاتصال بقاعدة البيانات من خارج المخدم ستبوء بالفشل ، لكن عملية محاولة الاتصال بقاعدة البيانات من داخل المخدم لا تحتاج للListener إذ أنه متخصص لاستقبال الطلبات الخارجية .

المتنصت Listener :هو إدارة تقوم بالاستماع على المنفذ Port لكي تستقبل طلبات المستخدمين من الأجهزة الأخرى على الشبكة الراغبين بفتح جلسة session مع الأوراكل سيرفر ... فالمتنصت ضروري جداً في حالة كون القاعدة منصبة على جهاز في شبكة و ترغب من الجميع الاتصال بها .

يستطيع مستمع واحد One Listener خدمة عدد من الDatabase Instances ، وهو فى الاصل عبار عن ملف يوجد فى المسار الاتي :

C:\oracle\product\10.1.0\Db_1\network\admin\listener.ora

يستطيع مدير قاعدة البيانات تحرير وتهيئة هذا الملف ليستقبل طلبات الاتصال بقاعدة البيانات ،



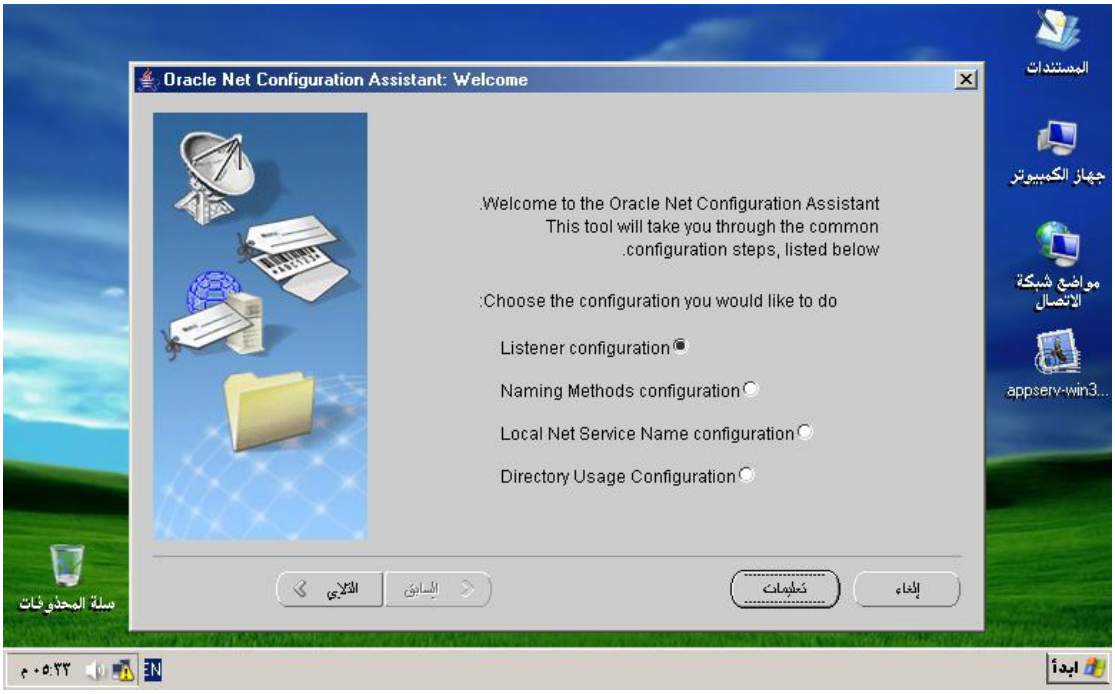
هذا نموذج لملف الListener بحيث يحتوى على One Listener يسمى Listener يعمل فى المخدم (SICOWIN اسم الجهاز) ويراقب الPort ١٥٢١ مستخدماً TCP PROTOCOL ، هذا المستمع يخدم INSTANCE تسمى ORCL.

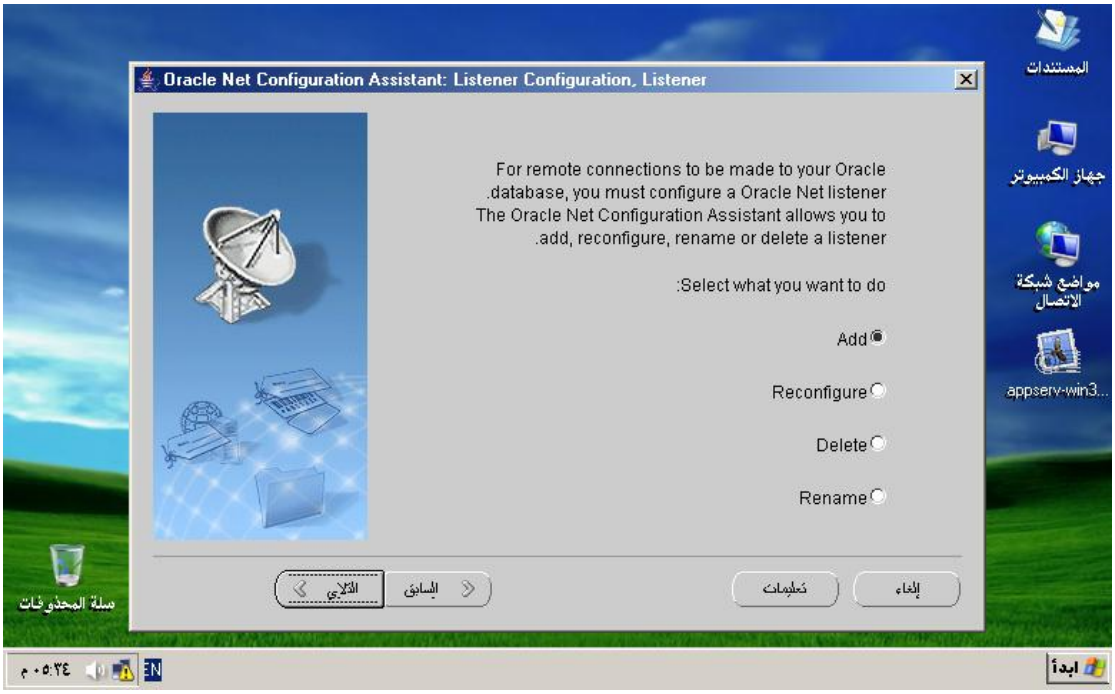
بالطبع يستطيع مدير قاعدة البيانات إضافة مستمع LISTENER جديد ولنفترض أننا نريد إضافة مستمع جديد يسمى LISTENER1 يعمل فى نفس المخدم SICOWIN يراقب الPORT ١٥٢١ ويخدم نفس الINSTANCE التى تسمى ORCL مستخدماً TCP PROTOCOL .

إضافة متنصت جديد :

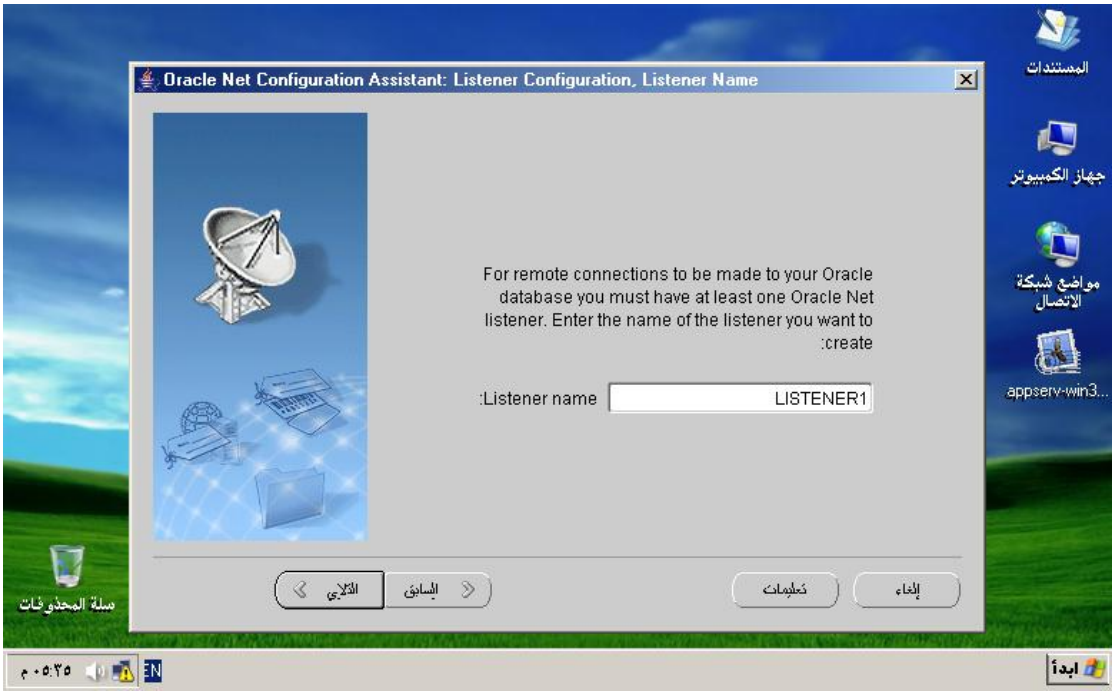
1-من Start->Programs->oracle10g->migration and configuration tools->Net Configuration Assistant

2-تظهر واجهة نختار منها Listener Configuration ثم .. Add

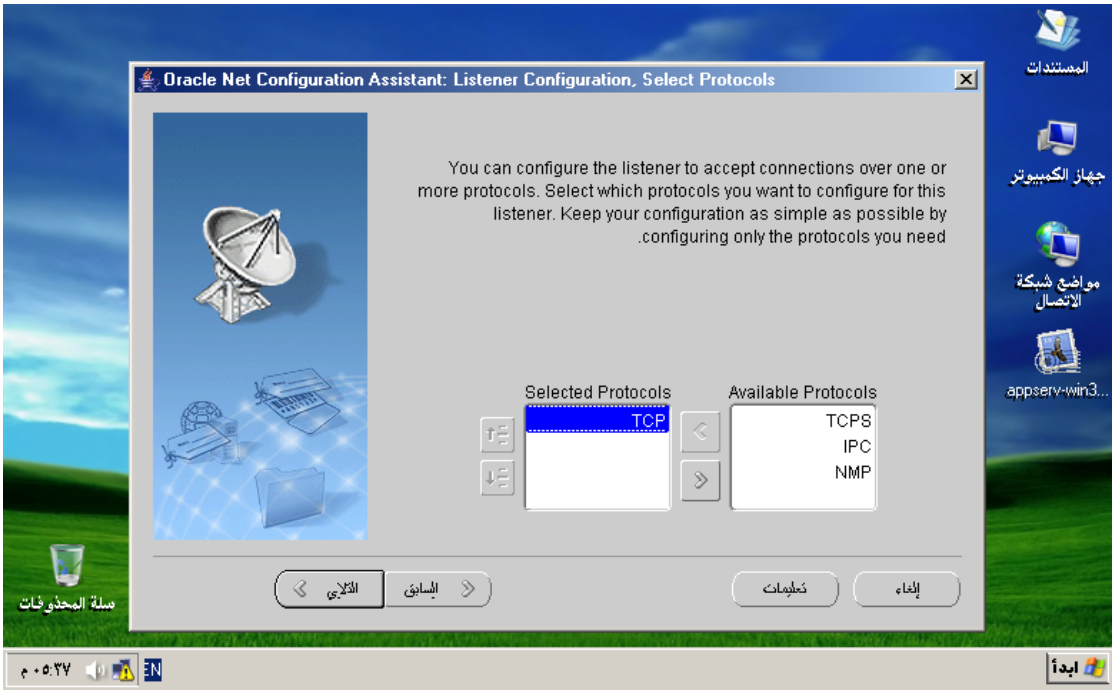




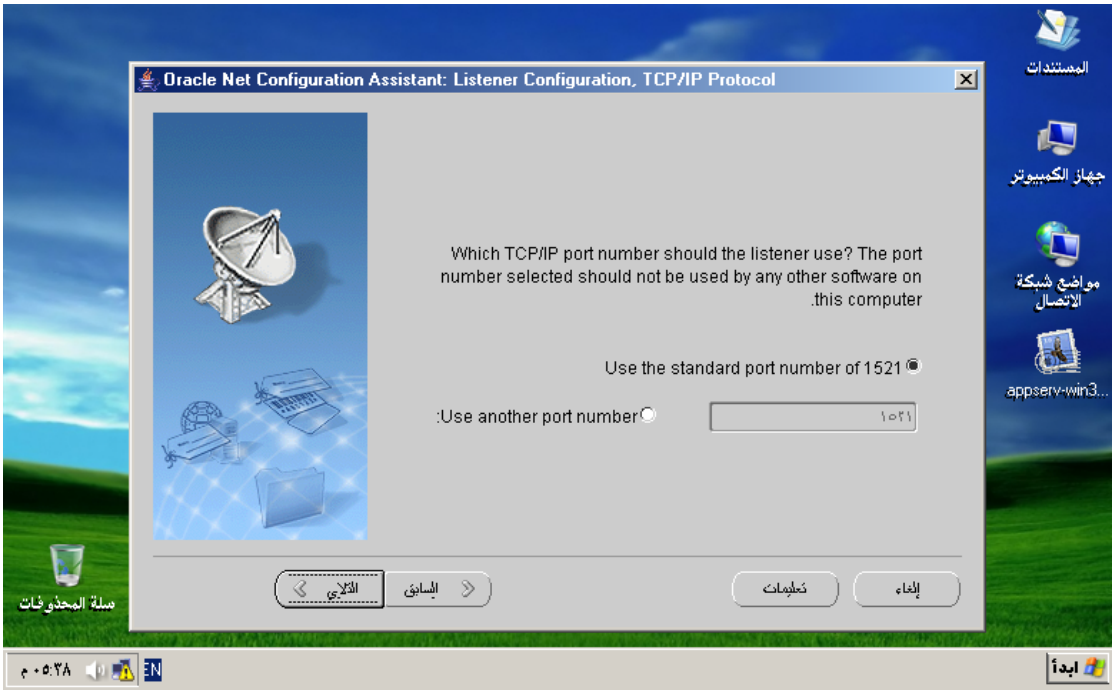
-3نضع اسم .. Listener



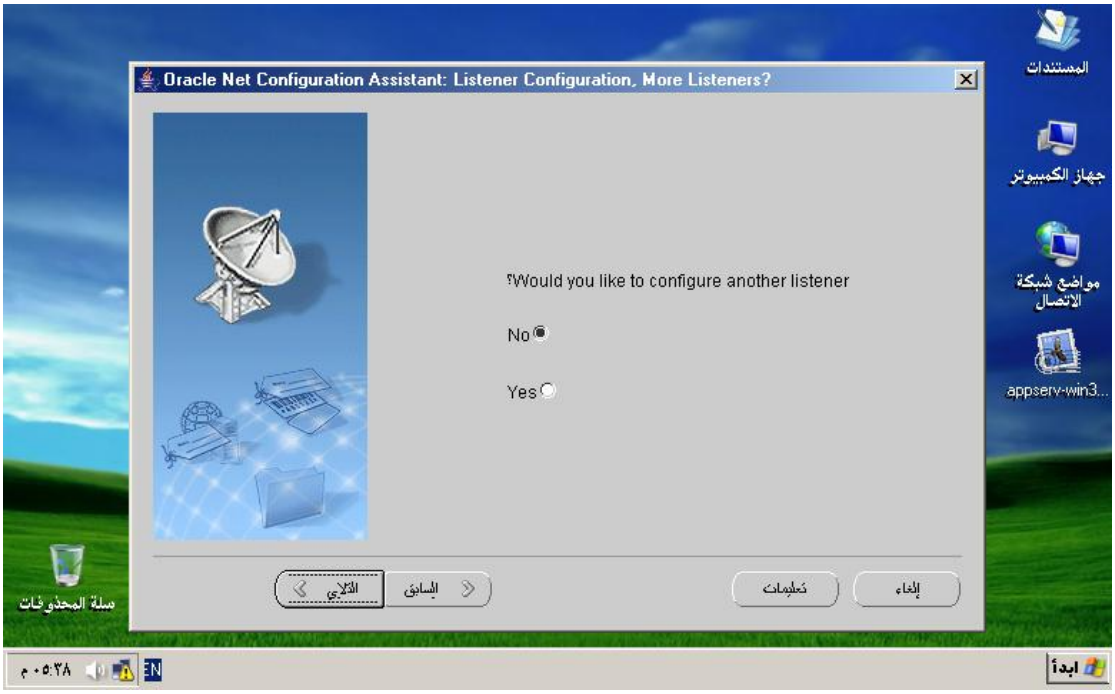
-4نضع البروتكول Protocolالذي نريد أن يستمع إليه المتنصت : في العادةً نختار tcp لشبكات الويندوز



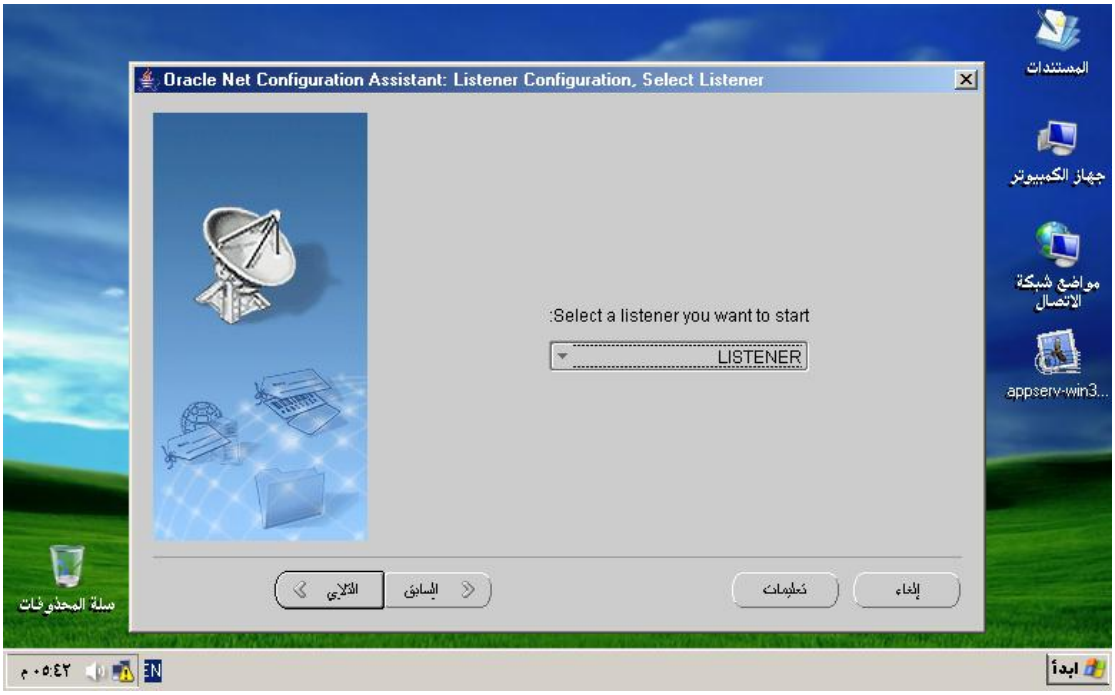
-5نختار port الذي نريد للمتنصت أن يعمل عليه ، عند اختياره يجب عليك سؤال مدير النظام لأن هذا المدير قد يكون قد أغلق Ports من أجل حماية السيرفر .. في العادة أختار منفذ أوراكل الافتراضي ١٥٢١ ...



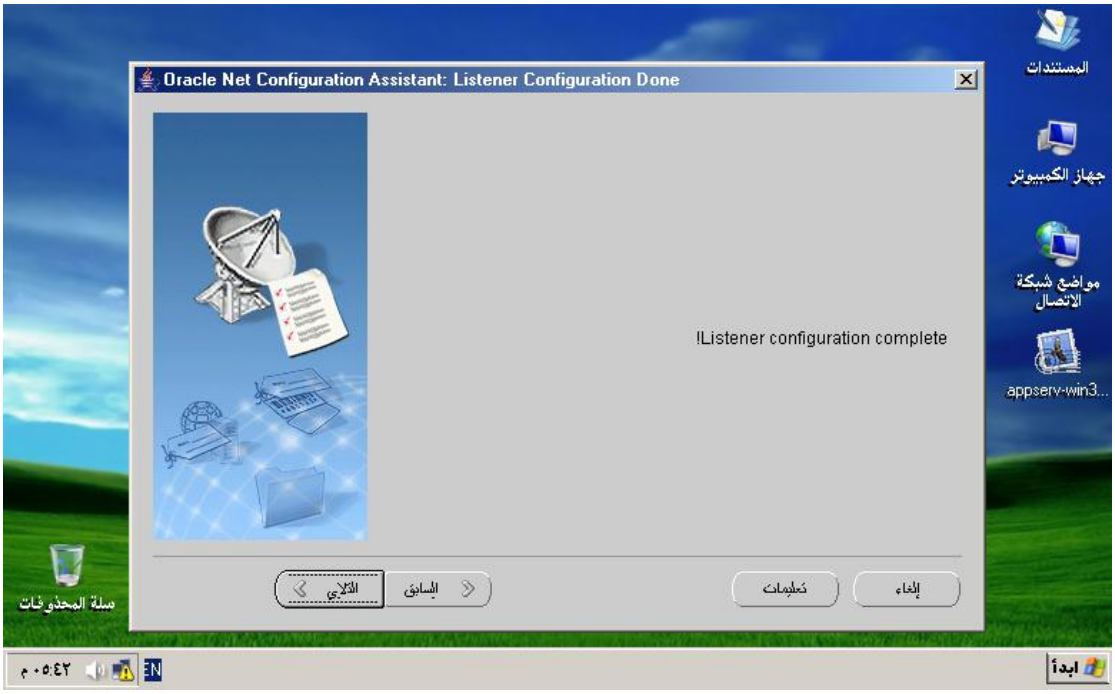
-5يسألنا هل نريد أن نقوم بإعداد متنصت آخر : نشكره على ذلك و نقول له لا ... ثم.. Next



-6يطلب منا هل نريد تشغيل متنصتنا الجديد .. يمكن أن نختاره ثم نضغط .. Next

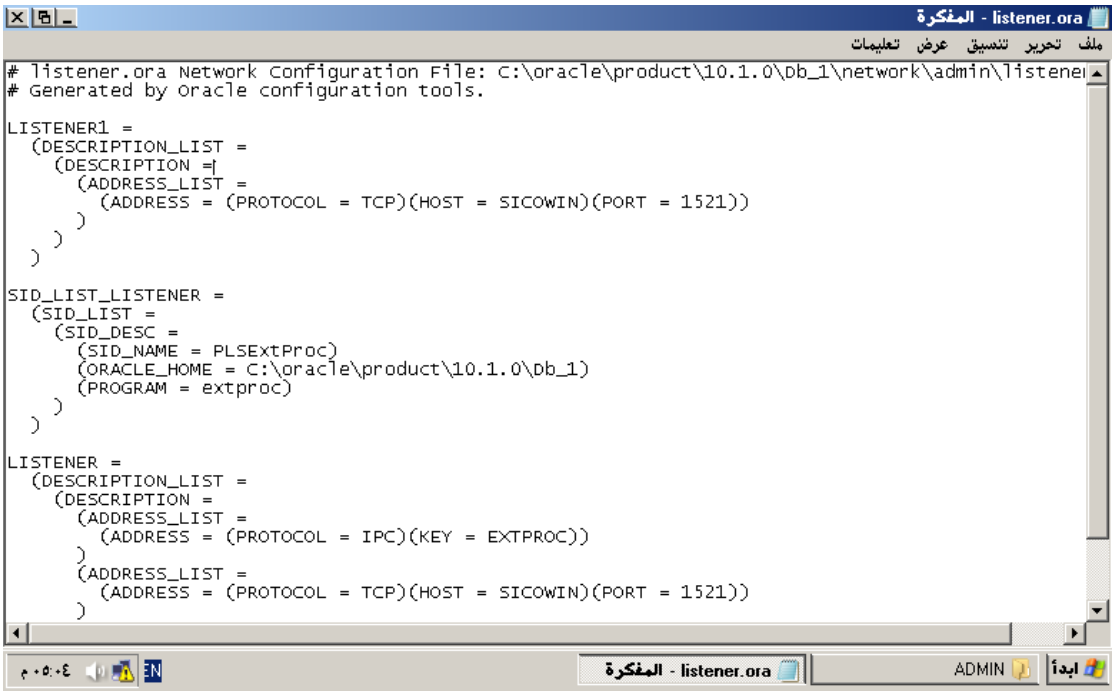


-7يظهر رسالة انتهاء العمل .... نضغط على التالي ثم نخرج من الواجهة ...



سهلة أليس ذلك ؟ ...

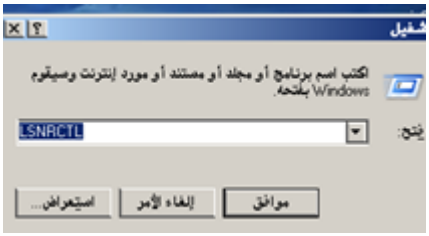
للعلم : فإن أوراكل عند تنصيبه فإنه ينشئ متنصب افتراضي يسميه Listener يتنصت على.. port : 1521

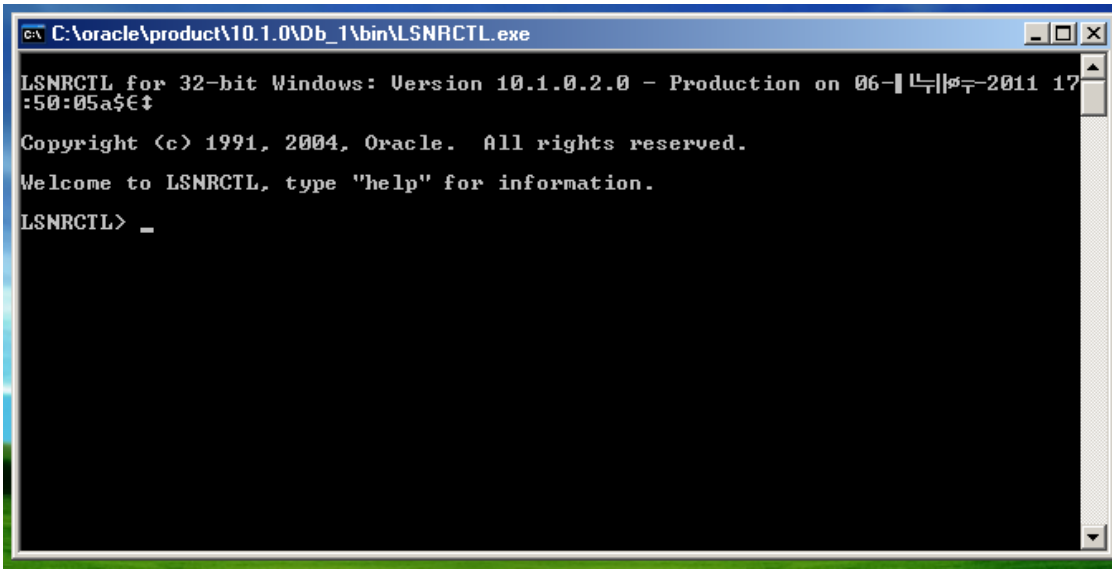


لاحظت التعديلات الجديدة فى الملف : وهى إضافة مستمع جديد يسمى LISTENER1.

بعد ذلك نستطيع التحكم فى LISTENERS بواسطة الامر LSNRCTL.

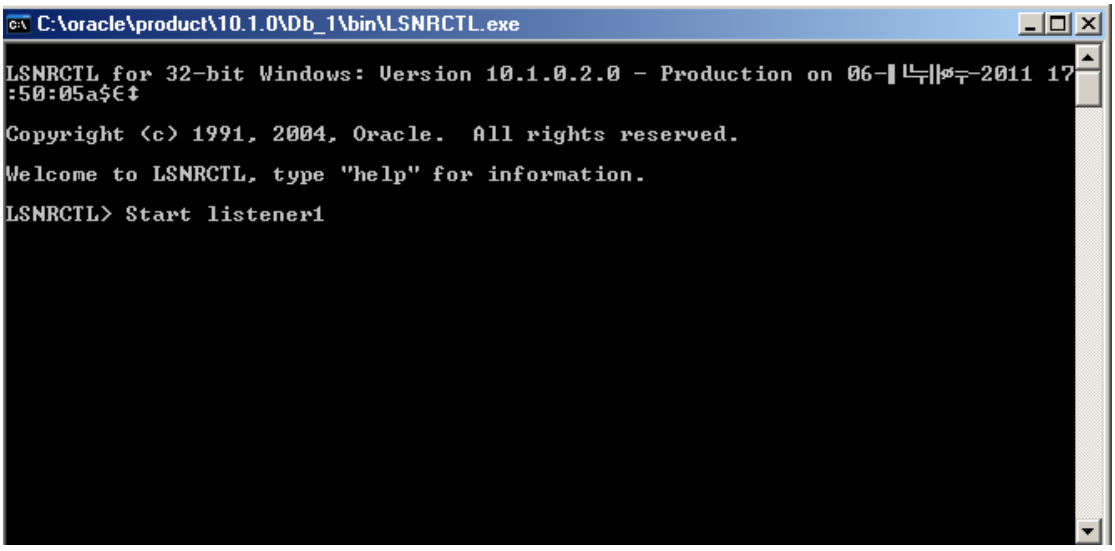
الاوامر المعروضة ه المتاحة للتعامل مع الLISTENER.

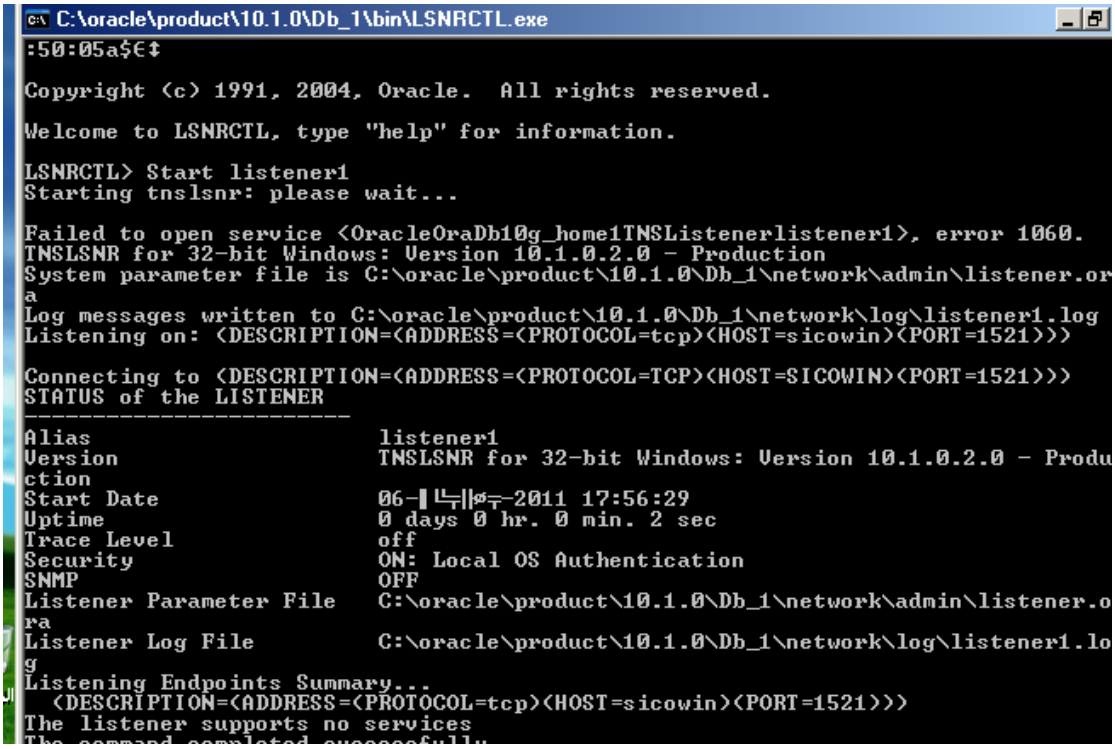




وهى اوامر لتشغيل وايقاف وعرض حالة المستمع وكذلك وضع كلمة سر وتغيرها للمستمع كنوع من التحقق ، وكذلك إعادة تشغيل المستمع ليستوعب ما تم تحديثه من تهيئة لملف الLISTERNER.ORA وغيره من الاوامر ، وبما أننا اضفنا مستمع جديد اسميناه LISTENER1 فالأفضل أن نقوم بتشغيل هذا المستمع ، لكن يجب التنبيه إلى أنه عند كتابة الامر LISTENER فإنه يتم التعامل مع المستمع الاصل أي DEFAULT والذى هنا هو LISTENER لذا إذا اردنا أن نتعامل مع المستمع غير الأصلي فأمامنا احد خياران:

الاول : تحيد اسم المستمع عند توجيه الامر



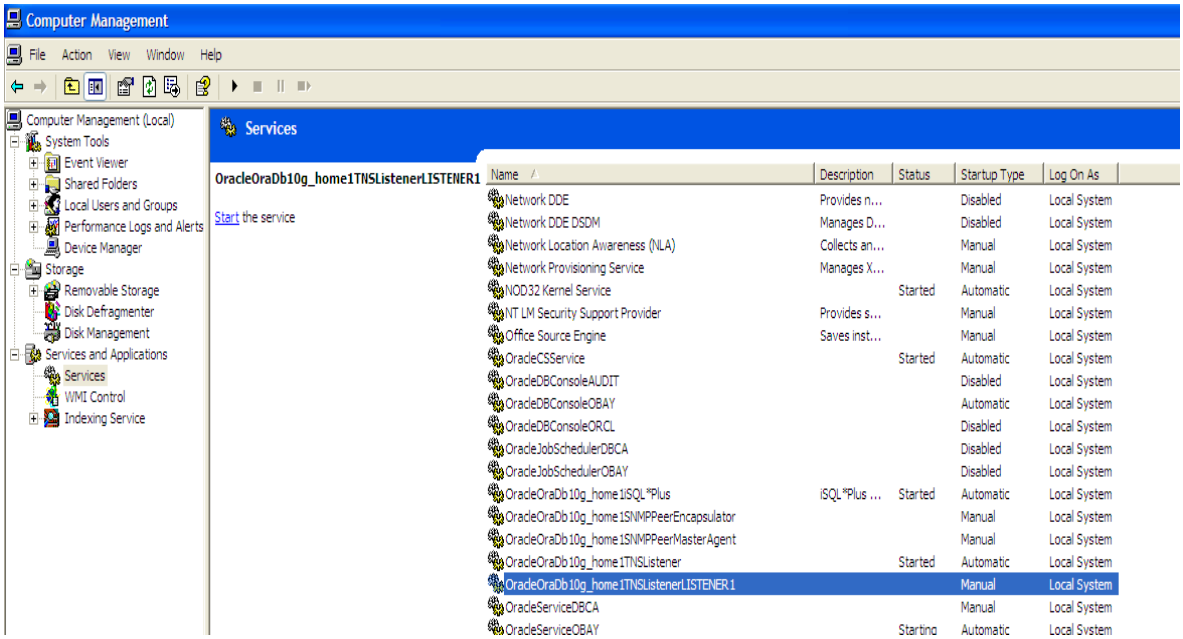


الان قمنا بتشغيل المستمع الجديد الذى انشأناه وهو المستمع LISTENER1 ، ولو لم نكتب اسم المستمع بعد الامر START لتم التعامل مع المستمع الأصلي وهو المستمع LISTENER.

الخيار الثاني : استعمال الامر SET CUR LISTENER_NAME لوضع الDEFAULT LISTENER الجديد ، أي سيصبح الاسم الجديد للمستمع هو الاصل الذى سنتعامل معه .

الان LISTENER1 هو الDEFAULT LISTENER.

بعد إنشاء المستمع LISTENER1 وتشغيله يمكن متابعته عن طريق الSERVICES



الان المستمع LISTENER1 يستطيع استقبال طلبات الاتصال بقاعدة البيانات على الPORT ١٥٢١ في المخدم NBS مستعملاً TCP PROTOCOL ليخدم ORCL INSTANCE.

لايقاف تشغيل

```
lsnrctl> Stop listener_name;
```

إذا أردنا مراقبة ومعرفة معلومات عن المستمع فالأفضل استخدام الامر STATUS او SERVICE لعرض معلومات عن المستمع مثل اسم المستمع واصداره ومتى تم تشغله والservices التي يخدمها ، ومسار ملف الLISTENER.ORA وغيرها من المعلومات.

كما ذكرنا أن مستمع واحد يستطيع خدمة عدد من الINSTANCES كما يمكن ان يتشارك عدد من المستمعين LISTENERS فى خدمة INSTANCE واحدة ، كما يمكن ان يكون هناك عدد من الINSTANCES فى الجهاز الواحد فيكون لكل واحد منها مستع يخدمها .

```
LSNRCTL> STATUS
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 10.1.0.2.0 - Produ
ction
Start Date                26-MAY-2008 23:57:51
Uptime                    0 days 0 hr. 17 min. 18 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   D:\oracle\product\10.1.0\db_1\network\admin\listener.o
ra
Listener Log File         D:\oracle\product\10.1.0\db_1\network\log\listener.log

Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROCipc)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=nbs)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=nbs)(PORT=8080))(Presentation=HTTP)(
Session=RAW))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=nbs)(PORT=2100))(Presentation=FTP)(S
ession=RAW))
Services Summary...
Service "OBAYXDB" has 1 instance(s).
  Instance "obay", status READY, has 1 handler(s) for this service...
Service "ORCL" has 1 instance(s).
  Instance "ORCL", status UNKNOWN, has 1 handler(s) for this service...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
Service "obay" has 1 instance(s).
  Instance "obay", status READY, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

من الإصدار Oracle8i فصاعداً يتم تسجيل الInstance الجديدة اليأ فى الDefault Listener أي لا يحتاج مدير قاعدة البيانات إضافتها يدوياً فى الListener ، وذلك فيما يعرف Dynamic Service registration .

سنتحدث عن الجانب الاخر وهو جانب الClients:

ليستطيع الClient الاتصال بقاعدة البيانات فى Database Server يحتاج لمعرفة بعض المعلومات الضرورية لنجاح عملية الاتصال :

١- المخدم او الHOST الذى يعمل فيه المستمع Listener.

٢- الPort الذى يراقبه المستمع.

٣- البروتكول Protocol الذى يستخدمه المستمع.

٤- اسم الservice او الInstance الذى يخدمه المستمع.

فعندما يطلب البرنامج او الApplication الاتصال بقاعدة البيانات من خلال المستمع Listener يحتاج لمعرفة المعلومات الموضحة اعلاه ليترجم ذلك فى عملية اتصال ناجح ، أما كيف يعالج هذه المعلومات لتتم عملية الاتصال فهناك عدة طرق :-

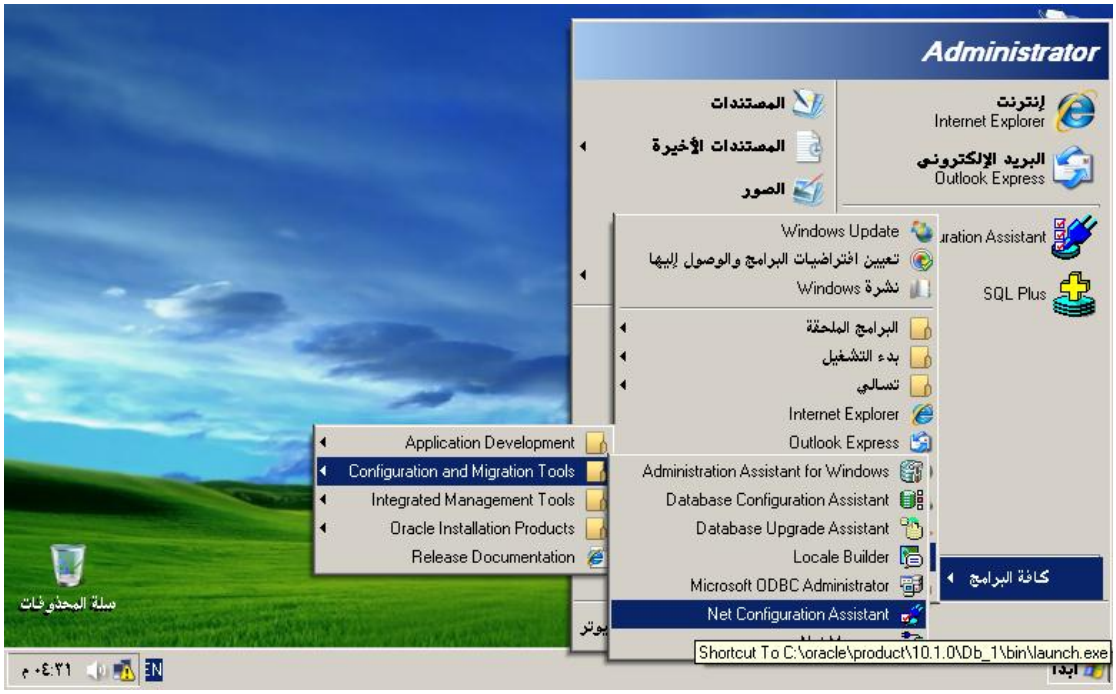
اتفقنا عليه أنه في أجهزة Clientنحتاج إلى تنصيب Oracle Clientو هذا ما تستطيع تنصيبه في نفس القرص الخاص بالأوراكل سيرفر Oracle Server سواء الإصدار ١٠ gأو حتى ٩i لن تجده في قرص (Oracle Database) ...

بعد تنصيبه يجب أن نخبر التطبيقات المختلفة أين ستوجد القاعدة ، و السؤال كيف ؟؟
يتم ذلك بواسطة ما يسمى Local Net Services Naming الذي نستخدمه لكي نحدد طرق الإتصال بالقاعدة

فعلياً فإن tnsnames.ora يحوي Local Net بالبنية التالية:

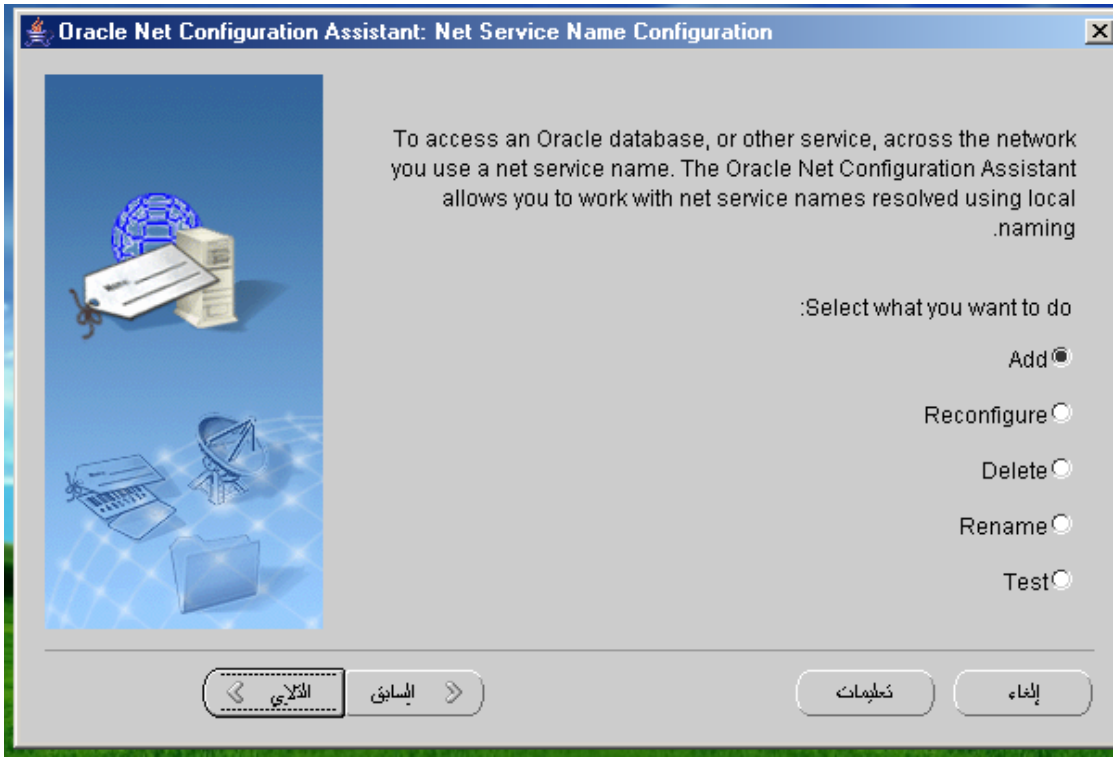
الآن لننتقل إلى الخطوات العملية التي يجب تنفيذها على أجهزة : client

1-نقوم بتشغيل ... Net Configuration Assistant



-2من الواجهة نختار كما في الصورة :

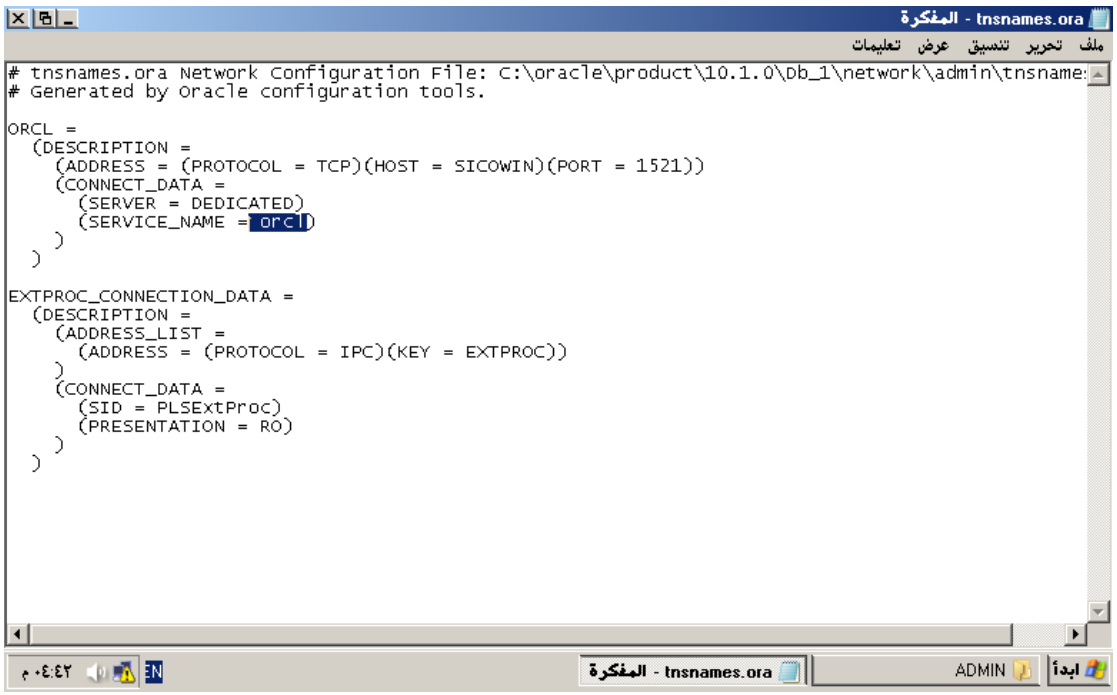
-3تظهر واجهة نختار منها . add

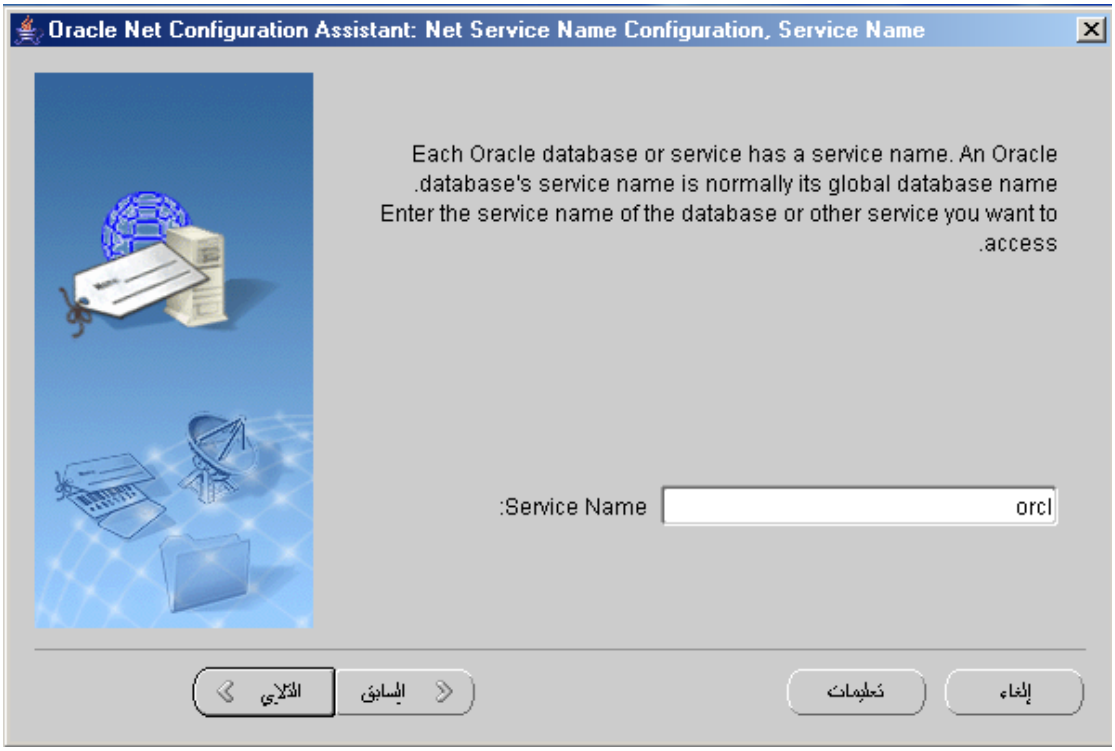


4-نضع اسم .. Service Name تتذكرون Global Name الذي وضعناه مع اسم SID اثناء انشاء قاعدة البيانات ؟؟؟ إذا لم تتذكرة اذهب الي المسار التالي في جهاز الخادم :

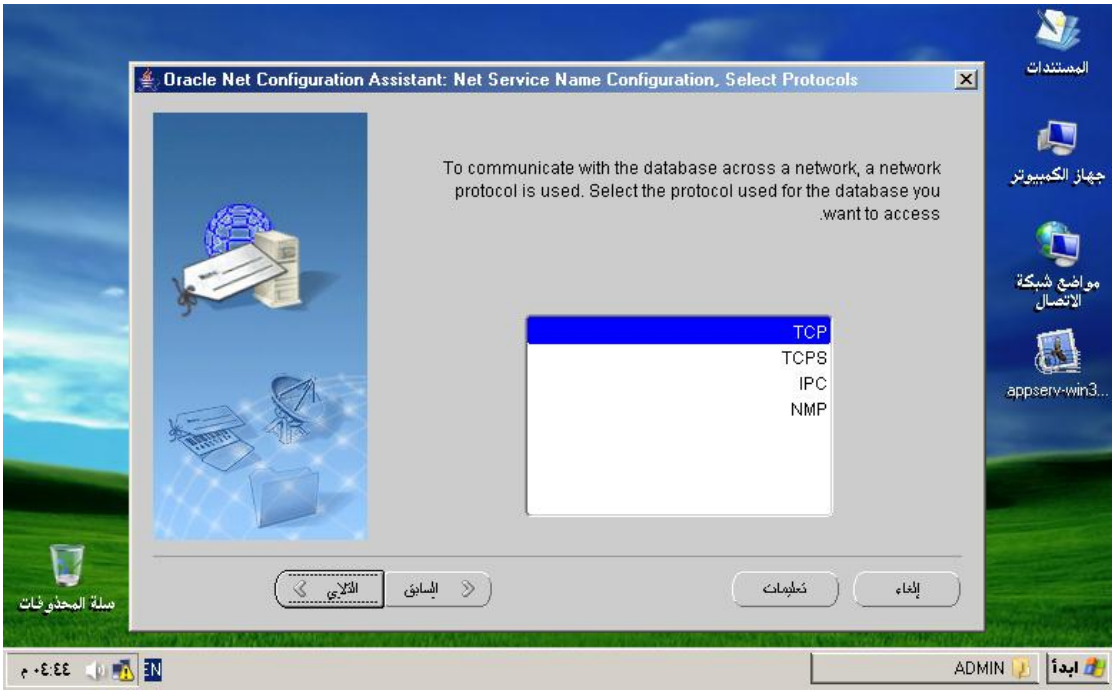
C:\oracle\product\10.1.0\Db_1\NETWORK\ADMIN \tnsnames.ora

ابحث عن ملف اسمة tnsnames.ora وفتحة يظهر كما في قاعدتنا كتالي:

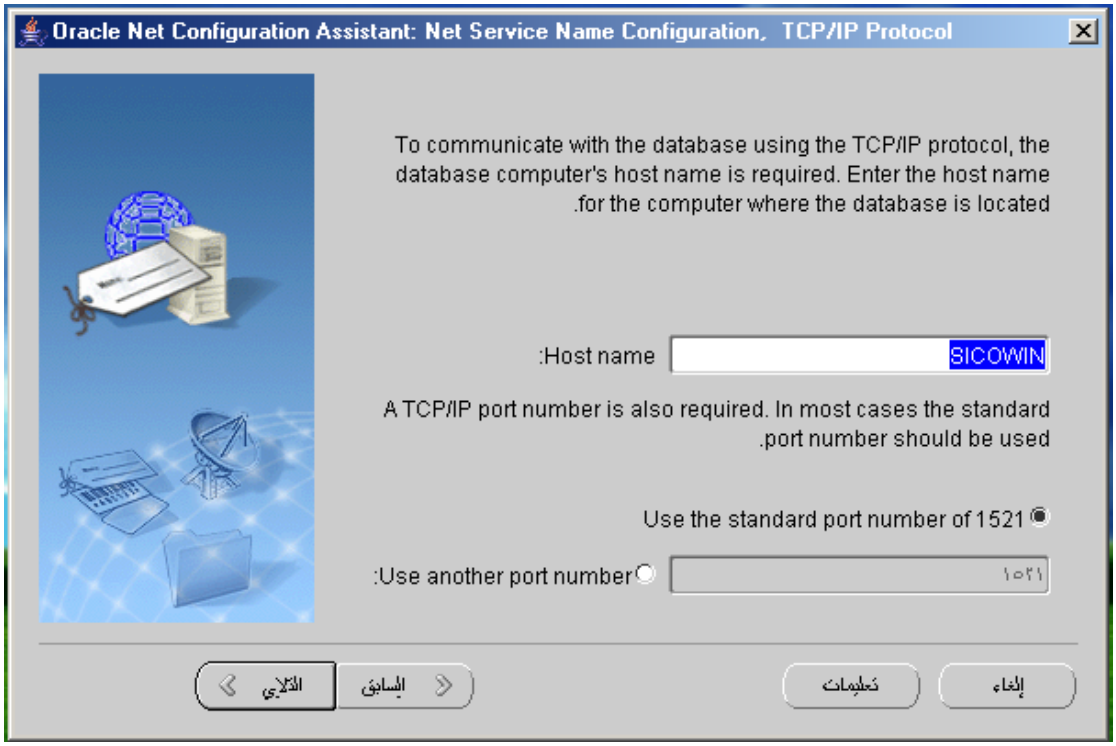




-5نضع اسم البرتوكول الذي سنعتمد عليه ... ضع .. TCP



-6ضع اسم الجهاز الذي يحوي القاعدة و رقم port الذي يعمل عليه المتنصت ...



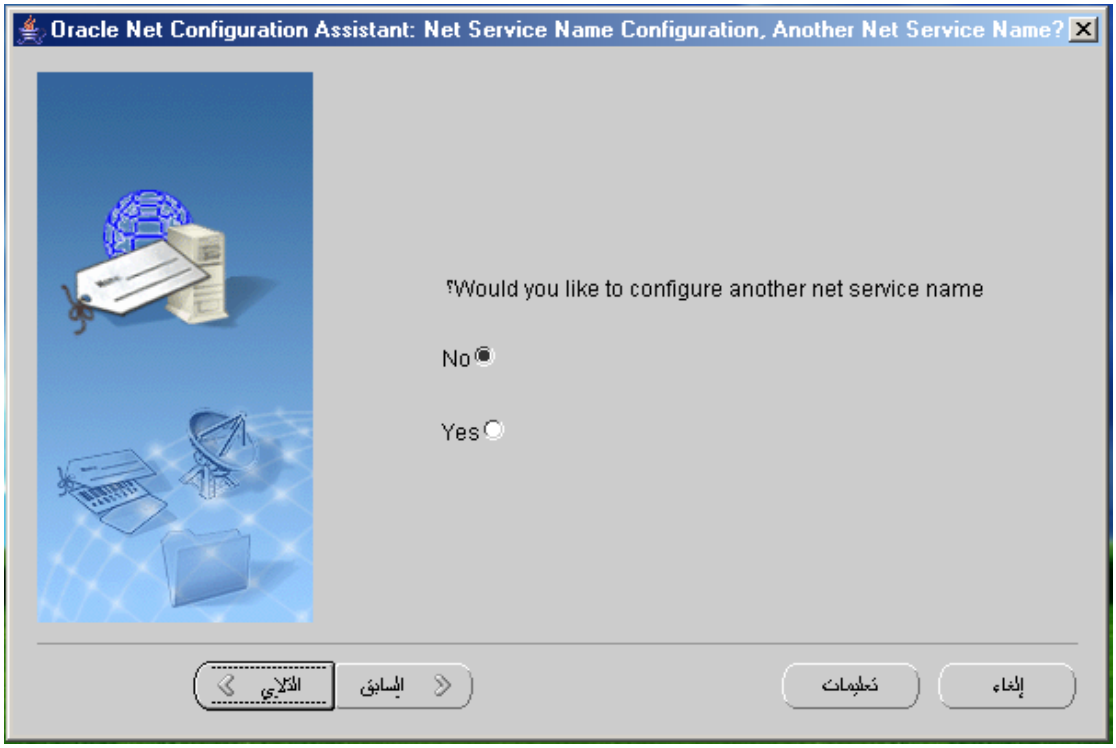
-7تظهر واجهة يسأل فيها هل نريد تجربة الاتصال .. طبعاً نختارYes

ثم تظهر واجهة تبين لنا نجاح الاتصال أم عدمه .. اضغط على زر change Login و ضع كلمة المرور لـ SYSأو System ثمOk

إذا نجح الاتصال فمبروك إذا لم ينجح تأكد من إعدادات الشبكة لديك ....المهم يجب أن تظهر الصورة التالية :

-عند الانتهاء يطلب منك اسم NET SERVICE NAME و هو الاسم الذي ستعامل معه مستخدموك في برامجهم أي عندما سيسجلون الدخول إلى برامج Sql Plus سيضعون في Connection String ما تضعه أنت في .... NET SERVICE NAME وصلت الفكرة ؟؟ ...

9- تظهر واجهة تسالك إذا كنت تريد تهيئة جديدة اختار no كما في الصورة التالية :



اختار التالي ثم انهاء .

٢- Easy Connect -:

فى هذا النوع من الإتصال يحتاج المستخدم لكتابة كل المعلومات التى يحتاجها الClient للاتصال بقاعدة البيانات ، يكتب هذه المعلومات اثناء عملية الإتصال وذلك على النحو التالى:-

<username>/<password>@<hostname>:<listener port>/<service name>

```
CONN SYSTEM/SYS@ SICOWIN:1521/orcl
```

حيث SICOWIN: هو اسم الذى يعمل عليه المستمع Listener.

١٥٢١: هو الPORT الذى يراقبه المستمع.

orcl: هو اسم الInstance التى يخدمها المستمع.

وهذا النوع هو اسهل طرق الإتصال من حيث أنه لا يحتاج لتهيئة فى الClient.

٢- Database Link:-

فى قاعدة البيانات الواحدة يستطيع المستخدمون منح صلاحيات على الكائنات التي يملكها لغيره من المستخدمين ، أما إذا كان لديك اكثر من قاعدة بيانات فمن المتعذر تعامل المستخدمين بين قاعدة بيانات واخرى ما لم يكن لدينا Database Link فهي الرابط بين قاعدة بيانات واخرى ، وقد تحتاج ذلك كثيراً اثناء عملك فقد يكون لديك اكثر من قاعدة بيانات تحتاج لربطها مع بعض .

ولنفترض أن لدينا قاعدة بيانات تسمى ORCL

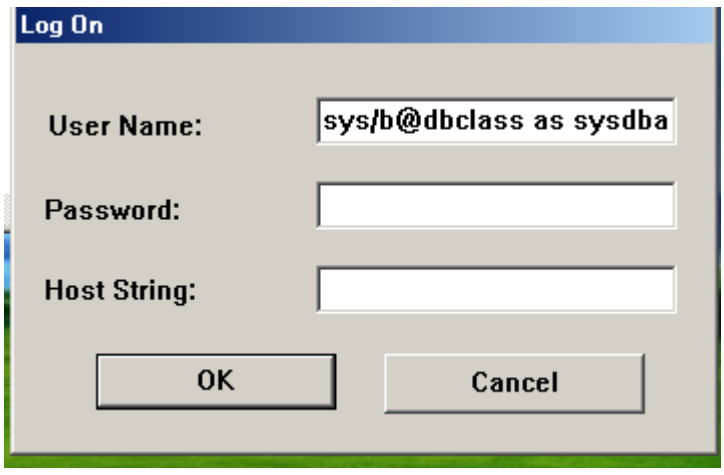
```
SELECT NAME FROM V$DATABASE;
```

واخرى تسمى dbclass.

نحتاج الان ربط قاعدة البيانات ORCL مع قاعدة البيانات dbclass ، ولنفترض أن المستخدم TEST فى قاعدة البيانات ORCL يحتاج لعمل استعلام على الجدول student المملوك للمستخدم VBS الموجود فى قاعدة البيانات dbclass ، إذاً نحتاج لعمل Database Link بين المستخدم TEST فى قاعدة البيانات ORCL وبين المستخدم VBS فى قاعدة البيانات dbclass.

الان نقوم بانشاء المستخدمين كلاً في قاعدة:

١- نبدأ المستخدم VBS في dbclass
٢- ١ للدخول نكتب :



١-٣ للتأكد من اسم قاعدة البيانات التي اشتغل فيها نكتب الامر :

```
SQL> select name from v$database;


NAME

---------

DBCLASS
```

١-٤ انشاء المستخدم

```
create user vbs identified by vbs

default tablespace users

quota 5 m on users;
```

١-٥ اعطاء المستخدم الصلاحيات التالية :

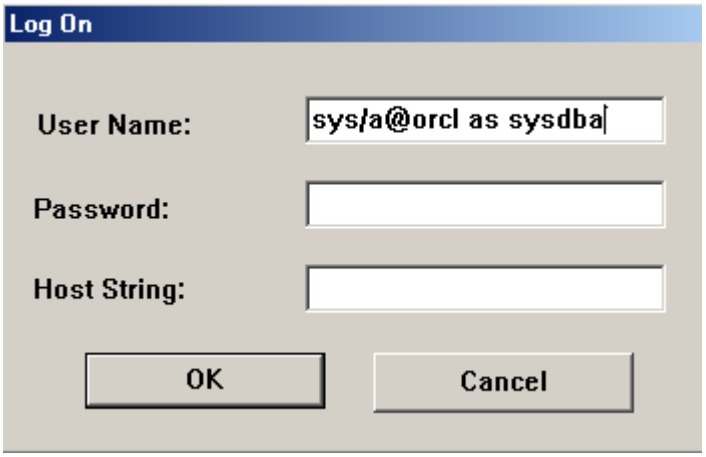
```
grant create session ,create table to vbs;

SQL> connect

Enter user-name: vbs/vbs@dbclass

.Connected
```

١-٦ انشاء جدول وتعبئته في البيانات

```
SQL> create table student(id number,name varchar2(50));

insert into student values (1,'salah');

insert into student values (2,'mona');

insert into student values (3,'wasf');
```

ثانياً : إنشاء المستخدم

٢-١ الدخول للمدير قاعدة البيانات :



٢-٢ للتأكد من اسم قاعدة البيانات :

```
SQL> select name from v$database;


NAME

---------

ORCL
```

٢-٣ انشاء المستخدم

```
create user test identified by test

default tablespace users

quota 5 m on users;
```

```
grant create session,create table,create database link to test;
```

للدخول للمستخدم نكتب:

```
SQL> connect

Enter user-name: test/test@orcl
```

لكن قبل إنشاء الDatabase Link لابد من الإشارة إلى أن المستخدم الذى يقوم بإنشاء ال Database Link لابد أن يكون لديه الصلاحية CREATE DATABASE LINK.

والان المستخدم TEST فى قاعدة البيانات ORCL سيقوم بعمل Database Link بينه وبين المستخدم VBS فى قاعدة البيانات dbclass.

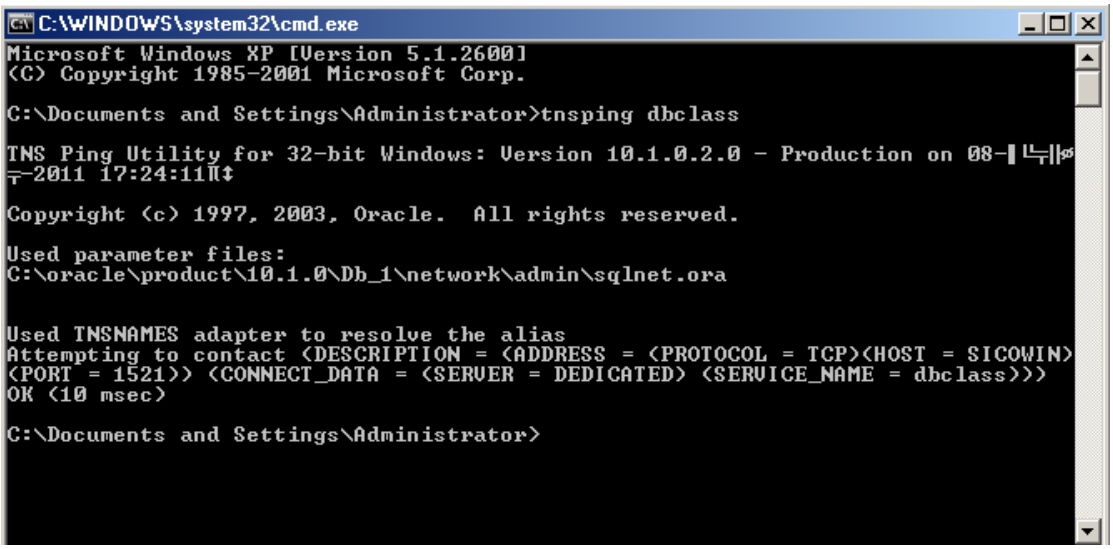
خطوات ربط بين القاعدتين :

اولاً: يتصل المستخدم TEST ويتأكد أن لديه الصلاحية CREATE DATABASE LINK.

```
SQL> SELECT * FROM USER_SYS_PRIVS WHERE PRIVILEGE='CREATE
'DATABASE LINK;

USERNAME                       PRIVILEGE                                ADM
------------------------------ ---------------------------------------- ---
TEST                           CREATE DATABASE LINK                     NO
```

فى الخطوة السابقة تأكدنا اولاً أننا نعمل على قاعدة البيانات ORCL وتأكدنا ثانياً أن المستخدم TEST يملك الصلاحية CREATE DATABASE LINK.

ثانياً: يقوم المستخدم TEST باختبار الOracle Net Service aliases الموجودة فى ملف الtnsnames.ora وذلك للتأكد من سلامة الاتصال بقاعدة البيانات الاخرى وهى هنا dbclass .



لقد تمت عملية الاختبار بنجاح.

ثالثاً يقوم المستخدم TEST بإنشاء الDatabase Link.

```
CREATE DATABASE LINK TESTVBS CONNECT TO VBS IDENTIFIED BY
VBS USING 'dbclass';
```

لقد قام المستخدم TEST بإنشاء Database Link تسمى TESTVBS يستطيع من خلالها المستخدم TEST الاتصال بالمستخدم VBS الموجود في قاعدة بيانات اخرى تسمى dbclass.

رابعاً : الان يمكن المستخدم TEST الاستعلام عن الجدول student المملوك للمستخدم VBS الموجود فى قاعدة البيانات dbclass.

```
SQL> select * from student@testvbs;

ID NAME
---------- ----------
salah ١
mona ٢
wasf ٣
```

فى بعد الاحيان يكون اسم الDatabase Link مصحوباً باسم الجدول طويل فمن الافضل إنشاء مرادف synonym لتسهيل واختصار كتابة العبارات خصوصاً تلك التى يتكرر كتابتها.

لكن يجب مراعاة أن المستخدم يحتاج للصلاحية CREATE SYNONYM لإنشاء المرادف.

```
SQL> conn
Enter user-name: sys/a@orcl as sysdba
.Connected
SQL> grant CREATE SYNONYM  to test;
```

ثم نعود الدخول للحساب المستخدم test وانشا مرادف:

```
SQL> CREATE SYNONYM stud FOR student@testvbs;
```

الان يمكن كتابة عبارة الاستعلام بالشكل الاتي:

```
SQL> select * from stud;

ID NAME
---------- ----------
salah ١
mona ٢
wasf ٣
```

تجدر الإشارة هنا بأن نقول انك تستطيع من خلال الDatabase Link إجراء عمليات الDML وهى (SELECT & INSERT & UPDATE &DELETE) لكن من المتعذر اجراء عمليات الDDL مثل CREATE & ALTER & DROP.

PUBLIC DATABASE LINK:

ما قمنا به فى الخطوات السابقة هو ربط المستخدم TEST فى قاعدة البيانات orcl مع المستخدم VBS فى قاعدة البيانات dbclass عن طريق DATABASE LINK تسمى TESTVBS.

لكن هل يستطيع أي مستخدم اخر فى قاعدة البيانات orcl استخدام TESTVBS DATABASE LINK ؟ الجواب وبكل سهولة لا ، وذلك لأن الDatabase Link التى قمنا بخلقها هى خاصة بالمستخدم TEST وليست لكل المستخدمين وهى تسمى PRIVATE DATABASE LINK.

ولكى يستطيع كل المستخدمون فى قاعدة البيانات orcl الاتصال بالمستخدم VBS فى قاعدة البيانات dbclass نحتاج لإنشاء PUBLIC DATABASE LINK.

بالطبع لإنشاء هذا النوع نحتاج للصلاحية CREATE PUBLIC DATABASE LINK.

ندخل الي مدير قاعدة بيانات orclr ونمنح المستخدم test هذه الصلاحية :

```
grant CREATE  PUBLIC  DATABASE  LINK to test;
```

ثم ندخل حساب المستخدم test ونكتب :

```
CREATE PUBLIC DATABASE LINK PUBTESTVBS CONNECT TO VBS
IDENTIFIED BY VBS USING 'dbclass';
```

قام مدير قاعدة البيانات بمنح المستخدم TEST الصلاحية CREATE PUBLIC DATABASE LINK ومن ثم قام المستخدم TEST بإنشاء PUBLIC DATABASE LINK تسمى PUBTESTVBS ، الان يستطيع جميع المستخدمين فى قاعدة البيانات orcl استخدم PUBTESTVBS DATABASE LINK.

للاستعلام عن الDATABASE LINK فى قاعدة البيانات

DBA_DB_LINKS

USER_DB_LINKS

ALL_DB_LINKS

V$DBLINK

بالطبع يمكن حذف الDATABASE LINK.

```
DROP DATABASE LINK TESTVBS;
```

كما يمكن حذف ال PUBLIC DATABASE LINK

```
DROP PUBLIC DATABASE LINK PUBTESTVBS;
```

بالطبع نحتاج للصلاحية DROP PUBLIC DATABASE LINK للحذف.

3- Materialized Views:-

عند مناقشتنا للDatabase Link عرفنا كيف يمكن ربط قاعدة بيانات بأخرى وكيف يمكن الوصول للكائنات فى قاعدة بيانات من قاعدة بيانات اخرى ، ولكن قد تحتاج فى بعض الاحيان لنقل وتحديث البيانات الموجودة فى قاعدة البيانات الى قاعدة بيانات أخرى .

كأن تكون مثلاً مديراً لمجموعة صيدليات ؛ كل صيدلية تحتوى على قاعدة بيانات ولكن على رأس كل ساعة تحتاج لجلب جميع حسابات الصيدليات من قواعد البيانات الموزعة الى قاعدة البيانات الرئيسية ، هذه هى مهمة الMaterialized Views .

سنستخدم هنا نفس المثال الذى استخدمناه فى الDatabase Link ، ولنفترض هنا أن قاعدة البيانات orcl هى قاعدة البيانات الرئيسية التى ستستقبل البيانات ، ولنفترض أننا سنستقبل البيانات فى المستخدم TEST ، أما قاعدة البيانات الاخرى dbclass والتى تحتوى على الجدول الرئيسي ولنفترض أنه EMPLOYEE المملوك للمستخدم VBS الذى نحتاج الى نقل وتحديث بياناته كل ثانية الى قاعدة البيانات الرئيسية orcl.

بالطبع نحتاج قبل كل شئ لعمل Database Link بين المستخدم TEST فى قاعدة البيانات orcl وبين المستخدم VBS فى قاعدة البيانات dbclass ، وذلك لعملية نقل وتحديث بيانات الجدول EMPLOYEE من قاعدة البيانات dbclass إلى قاعدة البيانات orcl.

أيضاً عملية تحديث البيانات (Refresh) بين الجدول الرئيسى والMaterialized Views تنقسم الى ثلاث أنواع :

١- FAST REFRESH:- فى هذا النوع من التحديث يتم فقط نقل البيانات التى تغيرت بعد اخر تحديث فهو لا يحتاج لنقل جميع البيانات ف الجدول وانما فقط ما تم تغيره بعد اخر تحديث ، هذا النوع بفى الغالب يختصر الزمن .

٢- COMPLETE REFRESH: أما فى هذا النوع من التحديث يقوم بنقل جميع بيانات الجدول من المصدر الى الMaterialized Views فيقوم بعمل إعادة كتابة البيانات القديمة واضافة البيانات الجديدة. ف الغالب ان هذا النوع يتطلب زمن اكثر من النوع FAST.

٣- FORCE REFRESH: هذا النوع بيدا اول بتطبيق النوع FAST إذا فشلت العملية كأن لم يجد مثلاً الMaterialized Views Logs فى جانب المصدر. ففي هذه الحالة يطبق النوع COMPLETE.

إذا لم يحدد نوع التحديث اثناء إنشاء الMaterialized Views فان الاصل DEFAULT هو FORCE.

-:Materialized Views Logs

كما سبق وان ذكرنا أن نوع التحديث FAST يقوم فقط بنقل البيانات التى تم تغيرها منذ اخر تحديث من الجدول المصدر الى الMaterialized Views. ولكن أين يتم تخزين معلومات عن البيانات التى تغيرت فى الجدول المصدر قبل نقلها الى الMataerialized Views ؟ الجواب يتم تخزينها فى ال Materialized View Logs ، وهو عبارة عن جدول يتم إنشاؤه فى قاعدة البيانات المستخدم الذى يملك الجدول المصدر وذلك عن طريق الامر التالي :

```
CREATE MATERIALIZED VIEW LOG ON EMPLOYEE;
```

بحيث EMPLOYEE هو الجدول المصدر.

فلحظة كتابة الامر اعلاه ؛ قاعدة البيانات تقوم بإنشاء جدول بالصيغة MLOG$_<TABLE_NAME>

:Primary Key Materialized Views

ولكن حتى تستطيع تنظيم الجدول المصدر ، دون التأثير على الFAST REFRESH وذلك بحيث يتم حفظ التغيرات بناءً على الPrimary Key ، فنستطيع لحظة إنشاء الMaterialized Views تحديد الخيار WITH PRIMARY KEY وهو الاصل DEFAULT فى حالة عدم تحديد خيار اخر ، والخيار الاخر هو ROWID.

لحظة إنشاء الMaterialized Views بالخيار Primary Key يجب أن يكون الجدول المصدر يحتوى على prmary Key Constraint.

كذلك عند تحديد الخيار Fast Refresh عند إنشاء الMaterialized Views يجب أن نكون قمنا بإنشاء الMaterialized View Logs فى المستخدم الذى يحوى الجدول المصدر وإلا ظهرت لنا رسالة خطأ.

قاعدة البيانات الاولى تسمى orcl ، سنقوم بإنشاء مستخدم فيها يسمى MAIN ؛ وسنقوم بإنشاء Materialized Views فى هذا المستخدم لجلب بيانات موجودة فى الجدول EMPLOYEE المملوك للمستخدم SUB الموجود فى قاعدة البيانات dbclass.

واليك الخطوات :-

١- سنقوم بإنشاء المستخدم sub الموجود فى قاعدة البيانات dbclass:

```
CREATE USER sub IDENTIFIED BY sub
Default tablespace users
quota 5 m on users ;
```

- ونقوم بإعطائه الصلاحيات التالية:

```
GRANT CONNECT,RESOURCE, CREATE MATERIALIZED VIEW TO sub;
```

- الدخول للحساب المستخدم sub و القيام بمايلي:

إنشاء جدول EMPLOYEE

```
Create table EMPLOYEE(emp_no number,emp_name varchar2(50),
CONSTRAINT  empno_pk PRIMARY KEY(emp_no));
Insert into  EMPLOYEE values(1,'ali');
Insert into  EMPLOYEE values(2,'omar');
Commit;
```

- نتأكد من الجدول المصدر وهو هنا EMPLOYEE والتأكد كذلك أنه يحتوى على PRIMARY KEY CONSTRAINT حتى نستطيع إنشاء MATERIALIZED VIEWS فى قاعدة البيانات الاخرى بإستخدام الخيار WITH PRIMARY KEY.

```
SELECT * FROM EMPLOYEE;
```

```
SELECT OWNER,CONSTRAINT_NAME,CONSTRAINT_TYPE,TABLE_NAME
FROM USER_CONSTRAINTS;
```

إذاً الجدول EMPLYEE يحتوى على حقلين ويحتوى كذلك على CONSTRAINT PRIMARY KEY.

- فى قاعدة البيانات dbclass وفى المستخدم SUB الذى يحوى الجدول المصدر EMPLOYEE نقوم بإنشاء MATERIALIZED VIEW LOG.

```
CREATE MATERIALIZED VIEW LOG ON EMPLOYEE;
```

قمنا بإنشاء MATERIALIZED VIEW LOG للجدول EMPLOYEE ويمكن التحقق من إنشاء الMATERIALIZED VIEW LOG للجدول EMPLOYEE بالاستعلام التالي:

```
SELECT * FROM TAB WHERE TNAME LIKE '%EMPLOYEE';
```

٢- سنقوم بإنشاء المستخدم MAIN فى قاعدة البيانات orcl وسنمنحه الصلاحيات الكافية.

```
CREATE USER MAIN IDENTIFIED BY MAIN
Default tablespace users
quota 5 m on users;
```

```
GRANT CONNECT,RESOURCE,CREATE DATABASE LINK,CREATE
MATERIALIZED VIEW TO MAIN;
```

٢- ١ الان فى قاعدة البيانات orcl نقوم بإنشاء DATABASE LINK بين المستخدم MAIN وبين المستخدم SUB فى قاعدة البيانات ORCL.

```
CREATE DATABASE LINK MAINSUB CONNECT TO SUB IDENTIFIED BY
SUB USING 'dbclass';
```

٢-٢- فى قاعدة البيانات orcl نقوم بإنشاء الMATERIALIZED VIEW.

```
CREATE MATERIALIZED VIEW EMPLOYEE_MV
   REFRESH FAST
START WITH SYSDATE
NEXT SYSDATE + 1/(24*60*60)
WITH PRIMARY KEY
AS SELECT * FROM EMPLOYEE@MAINSUB;
```

قمنا بإنشاء الMATERIALIZED VIEW مستخدمين الخيار FAST REFRESH وذلك لأننا

قمنا بإنشاء الMATERIALIZED VIEW LOG فى الجانب الاخر وإلا ظهرت رسالة خطأ ، وكذلك

استخدمنا الخيار WITH PRIMARY KEY وذلك لأن الجدول المصدر يحتوى على PRIMARY KEY CONSTRAINT ، على أي حال هناك خيار اخر هو WITH ROWID.

٣- الان يمكن الاستعلام عن الجدول المصدر والMATERIALIZED VIEW ستلاحظ أنه كل ثانية كما حددنا ذلك [SYSDATE +1/(24*60*60)] تكون النتيجة متطابقة بين الجدول المصدر EMPLOYEE وبين الMATERIALIZED VIEW وهى EMPLOYEE_MV.

في المستخدم sub

```
SELECT * FROM EMPLOYEE;
```

في المستخدم man

```
SELECT * FROM EMPLOYEE_MV;
```

4- الان لو قمنا بإضافة حقل جديد فى الجدول المصدر ثم بعد ذلك أعدنا عمليات الاستعلام اعلاه.

```
INSERT INTO EMPLOYEE VALUES (3,'kasm');
```

لاحظت معي إضافة حقل ثالث فى الجدول EMPLOYEE.

ستنعكس الإضافة فى الجدول المصدر EMPLOYEE على EMPLOYEE_MV MATERIALIZED VIEW بعد ثانية واحدة من الاضافة فى الجدول المصدر.

لو قمنا بعمل استعلام للجدول MLOG$_EMPLOYEE قبل عملية الREFRESH سنجد معلومات عن الحقول التي اضيفت بعد اخر عملية REFRESH ولكن بعد عملية الREFRESH تختفى المعلومات السابقة عن الجدول وفى انتظار الجديد.

يستطيع مدير قاعدة البيانات عمل استعلام عن الMATERIALIZED VIEWS التي تعمل فى قاعدة البيانات بواسطة الجدول

DBA_MVIEWS

```
Select owner ,mview_name ,refresh_mode,query from  DBA_MVIEWS;
```

بالطبع المستخدم MAIN يستطيع حذف الMATERIALIZED VIEW.

```
DROP MATERIALIZED VIEW EMPLOYEE_MV;
```

أدوات الإدارة في : Oracle10g

فعلاً يمكننا أن نقول أن شركة أوراكل قد خطت خطوة رائعة مع إصدارها ١٠ gبأن توجهت إلى مديري القواعد و أمدتهم بعدد من الأدوات التي سهلت العمل كثيراً ، فمثلاً كنت مضطر إلى أن أحفظ مجموعة ضخمة من التعليمات لإنشاء مكونات القاعدة مثل الفضاءات الجدولية و المقاطع segmentsو الإمتدادات .. Extents بالإضافة إلى تعليمات إنشاء المستخدمين و حفظ الصلاحيات ... إلخ لذلك فإن دليل المديرين كان لا يفارق حقيبتي ...

الآن : وفرت أواركل أداة هي Enterprise Manager التي طورتها عن الإصدار ٩ iو أصبحت هذه الأداة تعمل عبر مستعرض الانترنت بالاستفادة من مكونات Application Server

بالإضافة إلى Enterprise Manager ظهرت iSQL PLUS بأدائها المطور عن الإصدار ٩ ... i

بالإضافة إلى أدوات أخرى .

الآن لنعد إلى الأداتين لنتعلم كيف نقوم بتشغيلها و إغلاقها (ملاحظة كل التعليمات التالية يتم تنفيذها عبر محرر الأوامر Dos .

1- ISQL PLUS :

لتشغيل الأداة :

```
isqlplusctl start
```

لإيقافها :

```
isqlplusctl stop
```

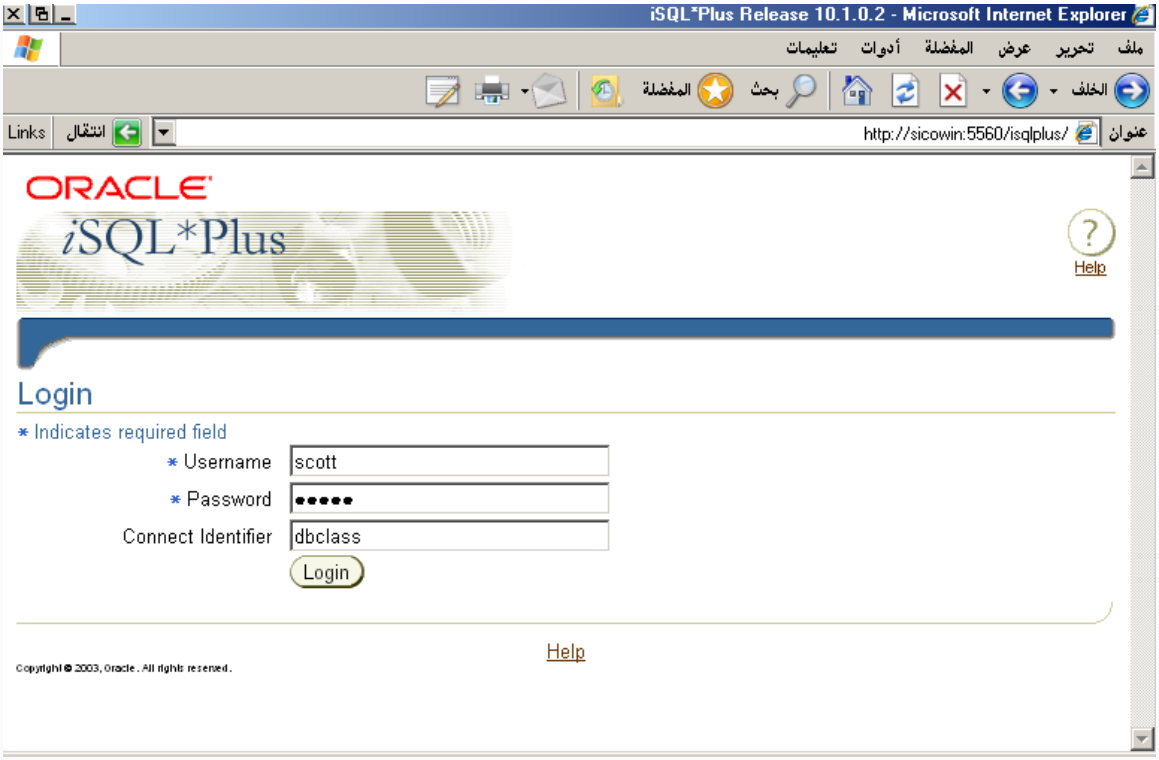
كيف نستعرضها :

يتم استعراضها عبر مستعرض الانترنت لذلك لابد من حفظ المسار اثناء الانتهاء من تنصيب اوراكل لانة يختلف من جهاز لأخر ، ولكن إذا نسيته اذهب الى المسار التالي :

C:\oracle\product\10.1.0\Db_1\install\readme

في حالتنا المسار هو:

```
http://SICOWIN:5560/isqlplus
```



Enterprise Manager -2

و هي الأداة التي نستطيع القيام بكل العمليات التي درسناها عبر مستعرض الانترنت بطريقة سهلة بدون كتابة الاكواد و إنما باستخدام المعالج الخاص بكل عملية وما عليك سواء اتباع خطوات المعالج.

لتشغيل الأداة :

```
emctl start dbconsole
```

لإيقافها :

```
emctl stop dbconsole
```

لإظهار حالتها :

```
emctl status dbconsole
```

لأخذ العلم لمحبي الويندوز :

الأداتين السابقتين يتم إنشاء Services خاصة بها تجدونها في Services داخل .. Administrative Tools و لا داعي لتشغيلها لأن تشغيلها يتم أتوماتيكي عند بدء التشغيل ..

لتشغيل الأداة عبر المستعرض : تماماً كما فعلنا مع : ISQL PLUS

```
http://SICOWIN:5500/em
```

إذا نسيته : يمكنك الذهاب إلى المسار التالي :

C:\oracle\product\10.1.0\Db_1\install\readme