

فهرست مطالب

۵.....	فصل اول
۵.....	معرفی پایگاه داده
۸.....	مفاهیم بنیادی پایگاه داده:
۹.....	نرمال سازی:
۱۱.....	نصب اوراکل express :
۱۳.....	برقراری connection به محیط client اوراکل
۱۶.....	نحوه ساخت table در اوراکل:
۱۸.....	فصل دوم
۱۸.....	Relation
۱۸.....	برقراری relation از لحاظ فیزیکی:
۱۹.....	مفهوم index:
۱۹.....	مفهوم profiles:
۲۰.....	ایجاد data file
۲۲.....	فصل سوم
۲۲.....	زبان SQL
۲۲.....	دستورات محیط SQL plus :
۲۷.....	ایجاد data file
۲۸.....	فصل چهارم
۲۸.....	Instance دیتابیس
۲۸.....	معرفی
۳۱.....	Instance

۳۱.....	Creating an oracle database
۳۲.....	ایجاد rule روی سطح database
۳۳.....	بازپس گیری نقش از کاربر (revoke)
۳۴.....	حذف نقش:
۳۵.....	قاعده NOT Null :
۳۶.....	Primary key
۳۷.....	قاعده check در database
۳۷.....	دستور describe
۳۷.....	دستور Data-Dictionary
۴۲.....	توابع
۴۷.....	ادغام دو جدول
۴۸.....	ایجاد جدول:
۴۹.....	ایجاد view برای جدول:
۵۱.....	فصل پنجم
۵۱.....	SQL
۵۱.....	Java- Core
۵۱.....	انواع متغیرها در جاوا:
۵۲.....	ساختار دستور switch
۵۲.....	حلقه ها:
۵۳.....	آرایه
۵۳.....	مفهوم کلاس:
۵۵.....	Interface

- ۵۶.....: کلاس انتزاعی
- ۶۰..... java native : Sql j
- ۶۲..... Data file ها:
- ۶۳..... ساخت repository :
- ۶۷..... oracle ساخت jar file در
- ۷۰..... انواع shut down در data base :
- ۷۴..... import, export و back up گیری
- ۷۹..... Batch file چیست؟

فصل اول

معرفی پایگاه داده

مقدمه

یکی از شرکت های بسیار قوی در زمینه نرم افزار، اوراکل است. جاوا محصول شرکت sun macro است و در حال حاضر شرکت اوراکل آن را خریداری کرده است از دیگر شرکت هایی که اوراکل آن را خریداری کرده است web logic, people sun, MySQL, می باشد. اوراکل یکی از سهامداران پایه شرکت تویوتا است.

مهم ترین محصول اوراکل پایگاه داده اوراکل (RDBMS) آن است. اوراکل علاوه بر RDBMS بودن ORDBMS هم شده است. دو تا از بزرگترین دیتابیس های دنیا اوراکل که یک RDBMS است و دیگری DB2 محصول شرکت IBM است و تفاوت های زیادی بین این دو تا دیتابیس وجود دارد. اما از مواردی که می توان از آن به عنوان تفاوت اوراکل و DB2 یاد کرد: اوراکل ذاتاً RDBMS است و قابلیت ORDBMS شدن دارد DB2 ذاتاً ORDBMS است که قابلیت RDBMS شدن را دارد.

هر RBMS روابط توسط کلید خارجی که معمولاً از نوع عددی است انجام می شود. Type فیلد Number=ID در ORDBMS، Type فیلد خارجی از نوع جدول مبدأ خواهد بود به این دیدگاه، دیدگاه object oriented نسبت به دیتابیس گویند یا همان دیدگاه شی گراء نسبت به دیتابیس. فیلدهای دیتابیس از نوع Data Type نیستند بلکه می توانند از نوع جدول دیگری باشند. این مفهوم را ORDBMS گویند.

رقیب اصلی اوراکل، DB2 است. MySQL هم در رتبه بندی های دیتابیسی وجود دارد ولی SQL Server به دلیل عدم رعایت برخی از قوانین و استانداردهای رتبه بندی دیتابیس نیست و رتبه اول و دوم معمولاً به اوراکل و DB2 تعلق دارد.

گران ترین دیتابیس جهان اوراکل است. و این نرم افزار در فاز lab مجانی و قابل دانلود از سایت oracle است ولی هنگام انجام پروژه های عظیم حتماً باید پشتیبانی (license) آن را خریداری کرد.

که سه مدل لایسنس داریم:

- عادی
- نرمال

- سیلبروگلد (در این نوع بیمه نامه ای به مشتری داده می شود که به ازای هر ۱۰ دقیقه ای که تراکنش روی DB اجرا شود از طرف شرکت اوراکل به مشتری خسارت پرداخت می شود).

(به دلیل تحریم ایران فقط از طریق کشورهای واسطه می تواند به لاینسس دسترسی داشته باشد که باعث مضاعف شدن قیمت می شود)

مشخصات خوب oracle (برتری های اوراکل نسبت به استاندارد DB)

نامحدود در تراکنش: تراکنش برای شما تعریف نمی کند. هر چقدر به اوراکل رم و CPU بدهید از آن استفاده می کند به شرطی که OS آن RAM و CPU را بشناسد البته در ورژن Enter prise آن.

برخی استانداردهای RDBMS:

- RDBMS می گوید که DB باید بر مبنای log اصلی کار کند که در SQL server این مقوله رعایت نمی شود و بر مبنای plate file کار می کند. در اوراکل رکوردها هم ذخیره نمی شود بلکه دستور SQL است که ذخیره می شود.

هر زمان که بخواهیم باید بتوانیم رکوردی را که delete کرده ایم برگردانیم (بدون Back up).

- تعاریف جامعیتی اطلاعات را پوشش بدهد، constraint ها را کامل در خودش داشته باشد (در SQL S رعایت شده است)

- زبان SQL را پوشش دهد (در SQL Server رعایت شده است).

مشکل SQL Server در این است که log به ما پس می دهد ولی روی plat file جستجو را انجام می دهد و اگر رکوردی را delete کنیم و بک آپ نداشته باشیم آن اطلاع را واقعاً از دست داده ایم ولی در oracle اگر رکوردی را از table حذف کنیم نه تنها table space کم نمی شود فضای اشغالی آن بیشتر هم می شود به علت attach شدن دستور delete و insert مگر اینکه اصطلاحاً flash کنیم یعنی insert با delete بر به یر از بین بروند. (مثل flash در hibernate).

اوراکل دارای سه تا فایل اصلی است یا سه تا فایل برای پروسیجروار کردن و یک سری table space دارد که ایجاد این فضای جدولی با کاربر است خاصیت یک RDBMS این است که با فایل log به صورت flat file برخورد نکند رکورد خام در فایل insert نشود که باید دستور SQL در کنار آن اضافه شود.

خاصیت دوم: می تواند چند مخزن را سرویس دهی کند (مخزن = repository). یعنی چند دیتابیس در آن داشته باشیم.

قابلیت های بحث های کلاسترینگ و وری هوس را کامل پوشش می دهد.

پشتیبانی از زبان PL/ SQL

PL/ SQL یک زبان جدید procedure base است و می توان دستورات SQL را به صورت function و procedure در سطح DB، store کرده و آنها را call کنیم.

پشتیبانی از زبانی به نام SQL که دقیقاً همان PL/ SQL است.

PL/ SQL ذات آن پاسکال است (ترکیب SQL + pascal) function oriented است.

SQL ← object oriented است و ذات این زبان، زبان Java است به خاطر همین ۹۰ درصد از کسانی که می خواهند oracle را تخصصی یاد بگیرند باید Java را یاد بگیرند بسیاری از مفاهیم در جاوا معنا پیدا می کند.

```
Public static dstning a( ) {
```

```
#
```

SQL دستورات

```
}
```

جاوا کار این application را از طریق JVM خودش run می کند و این کلاس در خود RDBMS، store می شود و به عنوان جاوا کلاس آن را store کرده و آن را رجیستر می کند a function را در application می توان call کرد.

+ اوراکل بسیار امن است البته در ورژن 11R2-، SQL از ورژن 8i به بعد ارائه شد.

برای نصب اوراکل حداقل باید 4Gig RAM داشته باشیم.

اوراکل دارای دو ویرایش در هر ورژن است.

- Express
- Enter price

تفاوت این دو در این است که در ویرایش Express، اوراکل بدون گرفتن lincens شما را پشتیبانی می کند ولی در Enterprise حتماً باید lisenss key داشته باشیم.

دلیل پشتیبانی رایگان در ورژن های Express:

تفاوت در برخی تعاریف

Express log

Express llj

- یک مخزن یا یک دیتابیس بیشتر نداریم
- بسیار سبک است
- محدود در تراکنش
- ۸ گیگ از رم را پشتیبانی می کنند
- بیشترین میزان ذخیره سازی اطلاعات در حدود ۴۰k است
- و در ازای اینها پشتیبانی آن رایگان است.

اما در oracle Enterprise تمامی خواص را پشتیبانی می کند و در واقع oracle express همان enterprise محدود است. هدف از تولید oracle express رقابت با My SQL (که به صورت open source عرضه می شد) بود. اما این مربوط به قبل از خریدن شرکت sun بود و با خریدن sun، صاحب My SQL هم شد.

مفاهیم بنیادی پایگاه داده:

Data: موجودیتی که دیتا را در خود ذخیره می کند.

Record: مجموعه ای از داده ها که کنار هم قرار می گیرند تا موجودیتی را توصیف کنند.

جدول: مجموعه ای از فیلدها که کنار هم قرار می گیرند.

در بانک اطلاعاتی مفهومی به نام Data Type داریم.

number = داده های عددی

char = کاراکتر، Varchar، nvarchar، n char

Text در oracle نداریم clob و nclob برای ذخیره سازی متن های وحشتناک طولانی.

به مجموعه ای از جداول که در کنار هم قرار بگیرند، یک schema گویند. Schema دقیقاً همان user است در oracle به user که table space داشته باشد schema گویند. یا همان user که مالک باشد. و هر schema مطمئناً user هست ولی هر user schema نیست. چون باید مالک یک object در سطح DB باشد تا بتوانیم آن را schema گوییم.

به مجموعه ی چند تا schema یک مخزن (دیتابیس) گویند. اگر چند تا مخزن داشته باشیم نیازمند نرم افزاری هستیم که آنها را پوشش دهد که اگر relational باشد RDBMS گویند و اگر object باشد ORDBMS گویند.

Instance: مخزنی که بالا می آید و آماده سرویس دهی می شود را گویند.

نرمال سازی:

در جدول روبرو در فیلد شهر دچار افزونگی اطلاعات شده ایم در نتیجه بهتر است برای جلوگیری از تکرار داده ها این جدول را نرمال سازی کنیم.

Id	name	Family	City
	amirsam	Bahador	Tehran
	Alireza	Shiri	Tabriz
	Hosan	taghipour	Tehran

پس آن را به دو جدول می شکنیم.

Shahrha			
Id- shahr		Name	
1		Tehran	
2		Tabriz	
Karmandan			
Id- shahr	Id	Name	Family
1	1	Amir	‘
2	2	Alireza	‘
1	3	Hosan	‘

تذکر:

هنگام برخورد به enti type جدید که دارای یک خصوصیت است را به یک جدول دیگر ببرید.

چند نوع رابطه در سطح Data Base داریم: دو نوع

۱. یک به یک

۲. یک به چند = چند به یک

- رابطه چند به چند را نرمال سازی می کنیم که از افزونگی داده جلوگیری کنیم.

در RDBMS یک سری نکات رعایت می شود.

مثلاً اگر رکوردی از جدولی حذف شود رکوردهای وابسته به آن هم حذف می شود.

تذکر: همواره آدرس فایل را به data base می دهیم نه خود فایل را، چرا که دیتابیس وظیفه انجام transaction ها و store کردن data های خام در

تراکنش های بالا را دارد.

Postgere ← RDBMS است.

کلید:

✓ کلید اصلی

✓ کلید خارجی یا فرعی

در oracle, Auto number نداریم بلکه sequence داریم. یعنی فیلد را از نوع number می گیریم و سپس جداگانه برایش یک sequence می نویسیم.

و sequence می تواند مشترک باشد بین چند جدول یعنی می توانیم unique key داشته باشیم بین چند تا جدول.

View: می تواند مشتق شود از یک یا چند جدول، حجمی از دیتابیس را می گیرد چون دستور SQL است ولی حجم رکوردی ندارد یک لایه است که

روی جدول قرار می گیرد از جدول یک خروجی می گیرد مثل فیلتر عمل می کند.

نام گذاری موجودیت ها: در سطح database یک استاندارد داریم که باید آن را رعایت کنیم:

✓ در نامگذاری نامی واضح برای موجودیت از جمله جدول فیلد و مخازن و وارد فرمائید.

✓ از وارد کردن space بپرهیزید و به جای آن از _ (underline) استفاده کنید.

- ✓ هیچگاه در ابتدای نام مورد نظر عدد استفاده نکنید.
- ✓ هنگام استفاده از DB حتماً caps lock را روشن کنید.
- ✓ دیتابیس به حروف بزرگ و کوچک حساس نیست ولی در مورد رکورد به حروف بزرگ و کوچک حساس است پس حساسیت فقط برای داده ها است نه برای نام جدول و فیلد و
- ✓ از کلمات رزرو شده استفاده نکنید.

دو نوع ارتباط با data base وجود دارد:

۱. ارتباط مستقیم: معمولاً این نوع ارتباط را admin ها دارند.
 ۲. ارتباط غیر مستقیم: کاربر از طریق application با data base تعامل می کند.
- در این کلاس نوع اول ارتباط بررسی می شود. اما می توان با خود RDBMS application تولید کرد با استفاده از نرم افزاری به نام apex که data base را از ورودی می گیرد و تم آن را می دهد. که web base است. بیشتر به درد مدیران می خورد. که apex, application builder در oracle express است که البته در express هم قابل نصب است.

نصب اوراکل express :

نصب oracle express 10g:

وارد فایل oracle express 10g می شویم. oracleXELriv را انتخاب می کنیم، عملیات set up شروع می شود. ابتدا RDBMS یا اوراکل سرور را نصب می کنیم بعد از آن باید اوراکل کلاینت را نصب کنیم و اصل اوراکل همان RDBMS است ولی محیط مدیریتی اوراکل در client است و این محیط مدیریتی در ورژن Express و Enterprise یکسان است.

با چند بار next به یک صفحه می رسیم که باید آدرس نصب اوراکل را بدهیم.

تذکر: هیچگاه اوراکل را در درایو سیستمی نصب نکنید.

تذکر: در مسیری که اوراکل را نصب می کنید اصلاً نباید space باشد.

در سربرگ security یک پسورد از ما خواسته می شود که در واقع پسورد تمامی admin ها خواهد بود. اوراکل وقتی نصب می شود چند تا admin دارد از جمله sys و system

پس از نصب RDBMS می دانیم که این سیستم دارای محیط SQL plus اوراکل است الان باید اوراکل کلاینت را نصب کنیم.

به فولدر oracle 10g s2bi4client می رویم. روی فایل set up کلیک نمی کنید روی set up راست کلیک می کنیم properties را باز می کنیم در سربارگ compatibility رفته.

Run this program ☒ تیک آن را می زنیم و موذ آن را windows XP(service pack2) قرار می دهیم. Ok را می زنیم حال به فولدر قبل برمی گردیم روی فایل set up دابل کلیک کرده yes را می زنیم.

که به محیط install oracle 10g می رویم next را می زنیم با path کاری نداریم. در قسمت destination در قسمت name می توانیم اسم دلخواهی بیاوریم که در program file فقط این اسم میاد و زیاد مهم نیست.

در path محل نصب اوراکل را تغییر می دهیم. Next می زنیم در صفحه بعد که select installation type است روی گزینه administrator کلیک می کنیم حال next را می زنیم در واقع در این صفحه انواع سرویس هایی که بر روی اوراکل وجود دارد را آورده است که کامل ترین آن Administrator است. install را می زنیم.

پنجره ای باز می شود که به چیزی دست نمی زنیم next را می زنیم دوباره می زنیم. در صفحه بعد نام سرویس را باید وارد کنیم ما اسم آن را XP می گذاریم. next ← tep ← next ←

Host name را local host می گذاریم که آدرس همان کامپیوتری است که client روی آن نصب است را قرار می دهیم. در قسمت پائین use the standard port 1521 ☒ را انتخاب می کنیم next می زنیم آیا می خواهید تست کنید yes را می زنیم. اگر در صفحه بعد پیغام invalid username & pass را نشان داد یعنی درست است و اگر ظاهر نشد یعنی ما به اوراکل وصل شدیم ولی user و pass اشتباه است. پس change login را می زنیم.

یکی از کاربردهای اوراکل system است آن را همراه با pass وارد می کنیم. Ok می کنیم حال می بینیم پیغام Successful connectivity ایجاد شده است.

در صفحه بعد net service name=XE بعد next می زنیم در این صفحه می پرسد سرویس دیگری می خواهید بسازی No را می زنیم. پشت سر هم next را می زنیم و سپس finish.

الان هم اوراکل سرور هم اوراکل کلاینت را به صورت جدا نصب کرده ایم.

حال می خواهیم به محیط مدیریتی اوراکل برویم طبیعتاً به start رفته روی oracle کلیک کرده سپس روی محیط Enterprise oracle consol کلیک راست کرده send to به دسکتاپ می کنیم. و shortcut آن در desktop ایجاد می شود. مشکل اینجاست که با کلیک روی آن محیط باز نمی شود.

می رویم به همان مسیری که آن را نصب کردیم به پوشه client رفته دکمه continue را می زنیم yes را می زنیم حالا دیگه می ریم از روی desktop کلیک می کنیم permission را به دست آورده ایم پس اجرا می شود. و دیگر Error نمی دهد.

حال در پنجره Add Database to tree در قسمت hostname می زنیم local host یا همان آدرسی که وارد کرده بودیم. SID همان اسم مخزن می باشد که آن را XE وارد می کنیم.

اوراکل یک bug دارد حرف اول keyboard را از روی text box ها نمی گیرد.

اسم XE اسم مخزن پیش فرض در اوراکل enter price است. و همچنین در آن هیچ مخزن دیگری نمی توانیم ایجاد کنیم. Ok را می زنیم اگر همه چیز درست پیش رفته باشد در tree مشاهده شده روی network و روی data base کلیک کرده XE-local host کلیک می کنیم از ما user و pass را می خواهد (system و pass) را وارد می کنیم. Ok را می زنیم حال روی data base, login داریم.

تذکر: در قسمت control panel ← change keyboard soy other ← location را از حالت Iron خارج کنید.

در قسمت فرمت هم English (US) بگذارید.

تذکر: در هنگام نصب اوراکل باید آنتی ویروس و firewall خاموش باشد.

تذکر: در اوراکل Express اگر در اواسط set up کامپیوتر ری ست بشود مشکلی پیش نمی یاد با روشن شدن سیستم از اول آن را نصب می کنیم. ولی در ورژن enter price در اواسط نصب برق قطع شد باید ویندوز را عوض کنیم اگر نصب آن کامل تمام شد راحت می توان آن را uninstall کرد.

می توان در navigator گزینه remove را بزنی تا دیتابیس اضافه شده به tree پاک شود و برای ایجاد مجدد DB روی add data base کلیک کنیم و همان اطلاعات را برای host name و SID و ... وارد می کنیم. حال روی local host که در tree هست کلیک می کنیم user و pass را وارد می کنیم و به DB وصل می شویم.

اگر دیتابیس در یک کامپیوتر دیگر باشد روی add data base to tree کلیک می کنیم. در host name آدرس آن سیستم را می نویسیم سپس repository آن را وارد می کنیم. همان XE دیگری است. با اینکار می ره روی آن سیستم و با مخزن XE آن رابطه برقرار می کند. حال باید user و pass ورود به آن را وارد کنیم.

اگر الان در یک سیستم دیگری به نام مثلاً Amir-PC وجود داشت و اوراکل سرور روی آن نصب بود ما از سیستم خودمان با استفاده از client می توانستیم از طریق شبکه به دیتابیس آن سیستم دسترسی پیدا کنیم.

برقراری connection به محیط client/اوراکل

ورود به نسخه client اوراکل

oracle → my oracle → enterprise manager →

در این محیط یک network داریم.

Navigator → Add data base to tree →

Host name: local host

→ ok → data base Port number: 1521 کلیک روی درختچه →

SID: XE

XE local host → $\begin{cases} \text{username: syetem} \\ \text{password: wjjwa123} \end{cases}$

همان pass که هنگامی نصب اوراکل وارد کردیم.

User های default در oracle: system و sys و sos cott

اگر همه چیز درست باشد درختچه دیگری باز خواهد شد که شامل instance و schema و security و storage و است.

ایجاد یک user جدید روی دیتابیس:

→ create → Security → users →

User و pass و ... ☐ Express password now کمک می کند که آیا این user همین حالا password ش expire شود یا نه. روی اولین login که می زنیم (user اگر تیک این گزینه را زده باشد) موظف است pass را عوض کند اگر آن را عوض نکند کاربر به حالت قفل خواهد رفت و دیگر کاربر بدون دخالت admin قدرت login به دیتابیس را ندارد. حتماً admin باید کاربر را از حالت قفل در بیاورد.

به تب role می رویم. بر روی DBA دابل کلیک می کنیم در قسمت granted تیک admin option که از این لحاظ برای ما اهمیت دارد: به ما کمک خواهد کرد که اگر amirsam که همان user کنونی است وارد سیستم شد قابلیت اعطای role ی که به خودش assign شده است را دارد یا نه.

پس تیک آن را می زنیم (در صورت تمایل) ← create

در ادامه آموزش ساخت user در همان پنجره create:

Authentication $\begin{cases} \text{password:} \\ \text{externod:} \end{cases}$

در این شرایط باید userی در OS خودمان ایجاد کنیم همنام با user اوراکل.

مثلاً یک user به نام system و ترجیحاً administrator هم وجود داشته باشد. بعد زمانی که به سیستم عامل login می کنیم به طور خودکار به دیتابیس اوراگل هم وصل شده ایم و دیگر اوراگل از ما user و pass نمی خواهد. Login در OS به منزله ی login در DB است. چند نکته: ۱- حتماً rule و admin سیستم را به این کاربر بدهیم.

در قسمت status: که locked یا unlocked می تواند باشد.

به صورت اتوماتیک وقتی Authentication:= Expire باشد کاربر قفل می شود.

ولی گاهی هم ممکن اس کاربر در حالت password باشد و lock شده باشد.

تب role :

به صورت پیش فرض در ورژن های مختلف اوراگل connect و rule هم به user assign می شود و کاربری که حداقل rule, connect را نداشته باشد بسیار مضحک خواهد بود. رول های دیگری هم به اسم delete و وجود دارد که رول های پیش فرض اوراگل هستند. اگر احساس کردیم که می خواهیم rule را خودمان بسازیم و به کاربری اعطا کنیم باید به تب system برویم. (گاهی دیده شده که کاربری rule, DBA را به user اعطا کرده و بعد به تب system برای جزئی کردن نقش آن می رود که این کار صد درصد اشتباه است چرا که rule, DBA کامل ترین نوع rule است.

در قسمت available برخی از گزینه ها را می بینیم:

Alter table ← تغییر دادن جداول

Alter DB ← تغییر دادن دیتابیس

Alter user ← تغییر دادن user

Create user ← ایجاد کاربر جدید

Create table ← ایجاد جدول

تب object:

در این قسمت می توانیم rule را به صورت موشکافانه به کاربر اعطا کنیم مثلاً روی فلان table فقط بتواند insert انجام دهد. یا روی column فلان جدول در user سیستم فقط بتواند رکورد یا data وارد کند.

اما انجام دادن اینکار به صورت تک تک برای همه کاربرها امری بسیار خارج از حوصله و در واقع اعصاب خوردکن است.

راه حل:

از قابلیت ساخت rule در اوراکل استفاده می کنیم مثلاً rule کارمندان را ایجاد می کنیم با محدودیت هایی که مد نظرمان است و زمان ساخت user این rule را به آن اعطا می کنیم.

Rule کلیک راست → → name rule → create role

هیچگاه توصیه نمی شود که در tab, role, رول های آماده ی oracle را به رولی که خودمان ساخته ایم assign کنیم.

object → System (تنظیمات روی جداول) create →

حال اگر با کاربری که ساخته ایم به database وصل شویم می بینیم که به error می خوریم چون محیط اینترپرایس قابلیت dynamic show ندارد یعنی ما به کاربرمان قابلیت تغییر فلان ستون از فلان جدول را داده بودیم و به دلیل عدم وجود dynamic show کاربر نمی تواند به سیستم وصل شود.

نحوه ساخت table در اوراکل:

Table حتماً باید یک مالک داشته باشد که اصطلاحاً به آن schema اطلاق می شود.

وارد schema می شویم user مورد نظر را پیدا می کنیم فرض کنیم قبلاً user با نام مهدی ساخته ایم اما الان به دلیل اینکه هنوز این کاربر مالک جدولی نیست آن را در قسمت schema می بینیم. پس روی هر کدام از schema هایی که به صورت پیش فرض وجود دارد (هر کدام که خواستیم) کلیک کرده روی قسمت table راست کلیک create را می زنیم.

حال در اینجا در قسمت schema نام user مهدی را پیدا می کنیم. و فیلدهای table را ایجاد می کنیم برای ورود اطلاعات به جدول ایجاد شده.

person → tables → MEHDI (نام جدول) راست کلیک edit content →

Apply را می زنیم و close را می زنیم (توجه اگر بر روی apply کلیک نکنیم و close را بزنیم data های ما save نمی شود).

یکی از مشکلاتی که وجود دارد این است که اگر ما داده های زیر را وارد جدول کنیم و apply را بزنیم آن داده ها در جدول ذخیره می شوند.

ID	name	family
1	Reza	Afshar
2	Leila	Mousavi
1	a	b

حال اگر مجدداً در ادامه وارد کنیم (1, a, b) آن را می پذیرد و این اشتباه است. pk نباید تکراری باشد.

راه حل:

در سربرگ constraint به آن اعلام کنیم که می خواهیم یک pk بسازیم در قسمت constrain definition می پرسد کدام فیلد را pk کنیم ID را انتخاب می کنیم apply را می زنیم و error می گیریم. چون ما همین حالا در جدول مقدار تکراری در ID داریم و قاعده را رعایت نکردیم.

همچنین در قسمت name می توانیم message مورد نظر به کاربر را وارد کنیم.

همچنین ما می توانیم با زدن تیک Null از اجازه خالی بودن فیلد را به کاربر بدهیم (هنگام design جدول)

در قسمت constraint می توانیم unique را هم انتخاب کنیم یعنی رکوردها نباید تکراری باشند.

(حروف کوچک و بزرگ روی دیتابیس تأثیر ندارد ولی روی رکوردها تأثیر دارد)

فصل دوم

Relation

Person

ID	Name	Family
1	Amirsam	Bahador
2	Ali	Rezai
3	Maryam	Latifi
4	Mojgan	Amiri

Person- Info

ID	ID- person	Adress
1	2	Tehran
2	1	Tabriz
3	3	Kish
4	4	Amol
5	4	۴ دو تا آدرس دارد Lavasun

برقراری *relation* از لحاظ فیزیکی:

به سربرج constraint از جدول person-Info می رویم و در قسمت type گزینه foreign key را انتخاب می کنیم ولی هنوز فعال نیست. باید در قسمت referenced schema، schema MEHDI را انتخاب کنیم و هم چنین در Referenced table باید person انتخاب شود (به عنوان جدول master) و در قسمت constraint definition، ID-person را به ID وصل می کنیم.

Apply را که بزنیم هنگام Data Entry دیگر مشکلی پیش نمی آید.

مفهوم *index*

دیتابیس برای اینکه بتواند روی رکوردها در جداول جستجو انجام دهد بر یک مبنایی باید جستجو را در بانک اطلاعاتی خودش انجام دهد. گاهی دیده شده که بر روی یک جدول جستجوی صورت می گیرد و در طول سه ثانیه تمام می شود و در همان جدول همان جستجو را انجام می دهیم یک روز طول می کشد. و این تفاوت به خاطر تفاوت در *index* گذاری می باشد. و در اونی که یک روز طول می کشد *index* گذاری مناسب انجام نشده است.

شاخص برای دیتابیس مفهوم است نه برای کاربرها فقط شاخص ها را به دیتابیس معرفی می کنیم و می گوئیم بهترین شاخص، شاخص ID است بهترین *index* برای جستجو است (شاخص های دیتابیس باید *unique* باشد دومین شاخص *national code* (کد ملی) است.

شاخص گذاری غلط باعث کند شدن سرعت می شود نه بالا رفتن سرعت.

اوراکل به صورت پیش فرض شاخص را بر روی فیلد *pk* ایجاد می کند. اگر هنگامی که *Query* می گیریم دیدیم کند عمل می کند شاخص مان را باید عوض کنیم تا به بهترین *performance* برسیم. شاخص بهتر است عددی باشد و بهتر است *sequence* باشد *Null* نباشد *unique* باشد یک زمانی ما بیشتر داریم روی یک شرایط خاص *Query* می گیریم مثلاً اکثر *Query* ها بر اساس شماره پرسنلی در یک جدول باشد حال در این جدول شاخص ها شماره پرسنلی است.

اولویت اول آن همیشه با *pk* است حتی اگر ما *pk* هم نداریم خود *oracle* برایمان یک *pk* می گذارد. ولی آن را به ما نشان نمی دهد.

و نکته عجیب این است که در اوراکل شاخص هم مالک دارد.

→ create → MEHDI → indexes - راست کلیک

در قسمت *table* اولویت بندی شاخص را مشخص می کنیم.

Create → ok

توجه: ضروری نیست که تمام فیلدها را شاخص اولویت بدهیم.

مفهوم *profiles*

Role قوی می کند *profiles* ضعیف می کند.

Schema → profiles → create →

Connect time ← مدت زمانی است که این profile به کاربر assign می شود یعنی مثلاً بعد از ۳۰ دقیقه کاربر به صورت اتوماتیک disconnect می شود. IDL time: یعنی اگر کاربر به همان میزان دقیقه روی دیتابیس تعامل نداشته باشد sign out می شود.

Concurrent session: دقیقاً محدودیتی است که اشک در می آورد، حداکثر چند تا session داشته باشیم: یعنی اگر با یک username و با یک password کاربرهایمان وارد database می شوند حداکثر می توانند مثلاً ۱۰ نفر باشند یعنی با system و My Java 123 فقط ۱۰ نفر بتواند remote کند.

همچنین اینکه چند block بتوانند روی session شان read کنند و چند تا block را بتوانند call کنند. (reads/session و reads/call)

سربرگ password:

Expire pass یعنی pass کاربر بعد از آن مدت زمان که تعیین کنیم، expire شود. و کاربر چند روز بعد از expire شدن قفل شود.

در قسمت history تعیین می کنیم تا چند تا از pass های قبلی کاربر نگه داری شود. و یا تا چند روز pass نگه داری شود.

.... Lock account on: بعد از چند بار login اشتباه کاربر به مدت چند روز قفل شود.

حال هنگام تعریف user هنگام انتخاب این profile را برای آن انتخاب می کنیم.

ایجاد data file

Table space:

روی table space در نوار ابزار سمت چپ کلیک راست می کنیم و گزینه create table space را انتخاب می کنیم و برای آن یک نام انتخاب می کنیم. حجم اولیه فایل را هم می توانیم اینجا مشخص کنیم و همچنین حجم اولیه را هم افزایش داد. در قسمت edit, file size (که حجم اولیه فایل است) را می توان تغییر داد.

Storage به صورت اتوماتیک increment است و مثلاً 5MG به 5MG می تواند رشد کند.

پس حجم اولیه آن 10MG است و وقتی آن 10MG استفاده شد 5MG به آن اضافه می شود و 15MG می شود. و تا بی نهایت می تواند hard را اشغال

کند یا تا زمانی که برسد به 100MGB که خودمان آن را تعیین می کنیم ← OK ← create

حال آن data file را در data space می بینیم

حال در قسمت security ← users ← راست کلیک create ← general ← نام (amirsam) و user و pass و مقدار default را table space و table space که به عنوان temporary محسوب می شود به این معنی که موقت است. یعنی اگر برای آن table space اول مشکلی پیش آمد از آن برای جایگزین استفاده می کند.

حال از این به بعد tableها و هر چیزی که Amirsam درست کند بر روی اون table space که الان براش تعریف کردیم می افتد. نکته مهم: ما می توانیم در آدرسی که در data file, properties, data file تعریف شده است خود data file را ببینیم.

هنگامی که روی database در قسمت instance, login می کنیم در قسمت session می توانیم بفهمیم چه کسانی که به oracle وصل شده اند. و می توان با کلیک راست روی آنها و زدن دکمه kill session سطح دسترسی آنها را از بین برد.

- هیچ وقت خود user و session هایی که توسط خود oracle در این قسمت ایجاد شده اند را kill نکنید.

فصل سوم

زبان SQL

زبانی بسیار ساده و مبتدیانه. بین تمامی بانک های اطلاعاتی دنیا مشترک است بجز SQL no.

Oracle دارای چند تا زبان است SQL و SQL و

زبان های SQL مختلف دارای dialect های مختص به خودشان هستند. دارای ساختار کلی و دستورات خیلی مشابه هستند. اما در برخی function ها با هم متفاوت هستند و SQL در oracle یکی از خواندنی ترین dialrct های دنیا است.

دستورات محیط *SQL plus* :

برای ورود به محیط SQL plus از محیط cmd استفاده می کنیم سپس باید user و pass وارد کنیم. اگر همه چی ok باشد در پائین آن پیغام connect می آید و >sql نوشته می شود.

- دستور show user :user می که با آن login کردیم را نشان می دهد.
- دستور set time on
- دستور set time off
- دستور host notepad;
- دستور help topic;
- دستور help host;

همه فیلدها از جدول T₁ را نشان بده.

select * from T₁

می توان این دستور را در دو خط بنویسید.

select *

From t.1

فیلد name را از جدول T₁ نشان بده.

select Name from T₁

مقدار دهی به جدول:

Insert into T1 (ID, Name) values (1, "amirsam");

اضافه کن به جدول T1 مقدار amirsam 1 را.

برای حذف مقادیر تکراری (رکوردهای تکراری) در یک ستون از دستور distinct استفاده می شود.

select Distinct Name from t2;

عمل حذف فیزیکی صورت نمی گیرد فقط در خروجی حذف می کند.

Select Distinct Name, family from t2;

فیلدها را بر اساس جفت فیلد Name و family, Distinct می کند پس اگر ستونی دو اسم مساوی ولی با فامیلی متفاوت داشته باشد هر دو آن در

خروجی نشان داده می شود.

select (ID-2) Havij, Name from t2;

عملگرهای * ÷ - + فقط برای اعداد به کار می روند و لاغیر.

شرط گذاری روی دستور select

select * from t1 where ID=2;

select * from t1 where ID=2 OR ID=3;

Id=2 AND AGE=26;

select * from t2 where Name != 'amir'; مخالف

select * from t2 where Name like 'amir Sam';

مانند مساوی عمل می کند و فقط و فقط مربوط به رشته است و مخالف آن NOT like است.

```
select * from t2 where Name like '---';
```

آنهايي که سه حرفی هستند.

```
like '—e---'
```

۵ حرفی هایی که حرف دوم آنها e است

```
like 'a%';
```

حرف اول آن a باشد و باقی حروف مهم نیست.

```
like '—a%';
```

حرف سوم آن a باشد و سایر حروف آن مهم نیست.

```
Like %e%;
```

در حروف وسط آن e باشد اول و آخر آن مهم نیست.

زمانی که می خواهیم دو فیلد را با هم OR کنیم از علامت II استفاده می کنیم و از علامت + استفاده می کنیم. اصطلاحاً به آن

```
select + (NAME II FAMILY) FullName from T2;
```

```
(NAME II ' ' II FAMILY)
```

```
DESCRIBE T2;
```

RowNum ستونی است که توسط oracle برای هر سطر شماره گذاری می کند که در select آن نمی یاد مگر اینکه صریحاً نام آن را بیاوریم و مقدار آن unique است.

```
select RowNum from t2; where rownum=3;
```

>

<

```
select * from t2 where rownum<3;
```

آنهایی که null هستند:

```
select * from t2 where family is null;
```

آنهایی که null نیستند:

```
select * from t2 where family is not null;
```

order by برای مرتب سازی

```
select * from t2 order by name;
```

ترکیب ورود داده ها را می توان از همین Row name بدست آورد.

Row name نسبی در خروجی رکورد نیست بلکه نسبی در ورود داده است، order by به معنای مرتب سازی است اگر از آن استفاده کنیم می بینیم که row name یک چیز فیزیکی ثابت است نه منطقی.

```
Delete from t2 where ID=2;
```

عملگر IN

```
select * from t2 where ID in (1,2,3,4);
```

آنهایی را که ID آن یکی از مقادیر (1,2,3,4) را دارا است.

عملگر between

```
select * from t2 where rownum between 1 and 20;
```

عملگر union

دو جدول مفروض است

```
select * from T2
```

```
select * from T3;
```

union فیلدهایی را که تکراری بوده اند را یکبار نوشته است.

```
select * from T2 union ALL select * from T3;
```


تکراری ها را هم نشان می دهد.

```
select * from T2 intersect select * from T3;
```

مشترک را نشان می دهد.

```
select * from T2 Minus select * from T3;
```

منها کردن

جدول دوم را از جدول اول کم می کند و خروجی را نشان می دهد.

(مقدار تابع) CHR اسم یک تابع است.

```
select CHR (ID) from T3;
```

کاراکترها را به صورت عکسی نشان می دهد.

```
Select CHR (65) Result from T3;
```

عدد را به کاراکتر تبدیل می کند و کاراکتر معادل آن عدد را برمی گرداند.

```
select concat (name, family) Result from T3;
```

```
Concat (concat ('Mr' , Name b'), ), family) Result from T3;
```

تابع SOUNDEX :

```
select soundex (Name) from T3;
```

این اندیس تلفظ نام است. A 400 ali (اندیس تلفظ Ali می شود A400)

حروف اول در اندیس می آید و باقی آن عددی می شود و این مناسب است برخی مواقع که نمی دانیم یک کلمه را چگونه می نویسند آن اندیس را search می کنیم.

```
select Name, family from T3 where soundex (Name)= SounDE x ('SEPIDE');
```

حالا کاربر علی را برمی گرداند و بر اساس تلفظ جستجو می کند.

ایجاد data file

در قسمت storage روی data file راست کلیک می کنیم create را می زنیم مسیر برای ساختن فایل به همراه نام آن را وارد می کنیم در قسمت size، size اولیه فایل را تعیین می کنیم y. در قسمت storage تیک Automatically extend را می زنیم، فرض کنیم سائز اولیه را ۱۰۰ انتخاب کردیم حال در تب storage در Increment اعلام می کنیم که مثلاً ۱۰ مگ، ۱۰ مگ به این فضا اضافه شود.

Unlimited یعنی اینقدر به size، 10MG، اضافه شود که hard کاملاً full شود.

یا اینکه می توانیم maximum size برای آن تعیین کنیم. مثلاً در value بزنییم 400M که تا 400M می تواند مدام به data file حافظه اختصاص دهد.

آدرس که data file در آن قرار است ساخته شود را کپی می کنیم دکمه create را می زنیم.

- به صورت پیش فرض در oracle نمی توانیم data file ایجاد کنیم که متعلق به table space ی نباشد.

به خاطر همین خود اوراکل یک table space پیش فرضی دارد به نام sys Aux که تمام data file های ما به صورت پیش فرض و اتوماتیک در آن table space اند بعد میتوانیم یگ table space خودمان تعریف کنیم و data file را روی آن قرار دهیم که اصطلاحاً آن را temp table space می کنیم برای اینکه ببینیم ok شده است در start آدرس را past می کنیم تا به آدرس آن temp table space بروود در فولدر XE روی SSS.ORA کلیک راست می کنیم در properties آن size آن را می بینیم 100M که تعریف کرده بودیم است.

حال اگر این 100M پر شود 10M به آن اضافه می شود و سائز آن 110MB می شود و محدودیت آن تا 400MB است.

- پسوند اسم فایل مهم نیست.

بانک اطلاعاتی شامل دو واحد اساسی است.

ساختار فیزیکی بانک: شامل یک سری فایل های فیزیکی بانک است.

فصل چهارم

Instance دیتابیس

معرفی

ساختاری که اجازه دسترسی به فایل‌ها فیزیکی همراه با یک سری سرویس می‌دهد.

می‌دانیم اگر بانک اطلاعاتی را در داخل یک سری فایل در نظر بگیریم برای دسترسی به فایل‌ها نیاز داریم که آنها cache شده به RAM ببایند لایه‌ای که دسترسی کار به این فایل‌ها را به ما می‌دهد همان instance است که شامل یک سری سرویس است که در داخل خود oracle موجود است مثلاً

ممکن است یک instance و یک ساختار فیزیکی داشته باشیم.

ممکن است چند instance و چند ساختار فیزیکی داشته باشیم که اصطلاحاً به این حالت Rac یا Real application Chester گویند.

پس زمانی که صحبت از Rac می‌کنیم یعنی با چند instance سر و کار داریم.

- تا زمانی که instance بالا نیاد database فعال نخواهد بود این instance به ما کمک می‌کند که با ساختار فیزیکی database بتوانیم تعامل داشته باشیم.

در Rac، ممکن است دو تا ساختار فیزیکی و دو تا instance یا یک ساختار فیزیکی دو instance و یا یک instance که دو تا ساختار فیزیکی داشته باشد داریم.

آیا file data ها را می‌توان به صورت مستقیم در OS باز کرد؟ خیر در اصل وظیفه instance است که ساختار فیزیکی را در اختیار ما قرار دهد. و معمولاً پسوند این فایل‌ها dbf (database file) است.

خود dbf ها از data block تشکیل شده‌اند و مجموع چند data block را extend و مجموعه چند تا extend یک segment نامیده می‌شود و مجموعه‌ای از segment ها را یک data file می‌نامیم.

نحوه خواندن data block ها توسط instance می تواند مضربی از ۲ باشد یعنی 2k یا 4k یا 8k یا 10k آنها را بخواند. وقتی instance 2k, 2k بخواند reference بر روی data file بیشتر شده و CPU بیشتر درگیر می شود وقتی عدد بزرگ تر می شود reference کمتر خواهد شد و CPU کمتر درگیر شد ولی در عوض RAM بیشتر مصرف می شود. این میزان را معمولاً در یک سیستم نرمال معمولاً تغییر نمی دهیم یعنی اگر نخواهیم روی linux و oracle نخواهیم کار کنیم آن را تغییر می دهیم.

Oracle بسیار بسیار database Manual است یعنی در آن همه چیز قابل تغییر است (به صورت دستی)

جداول در کجا ذخیره می شوند؟ در data file ها جداول و سایر ساختارها ذخیره می شوند.

از این رو چیزهایی مثل جداول (چیزهایی که در سطح دیتابیس size اشغال می کنند) اصطلاحاً object هایی از schema که در database size اشغال می کنند را گاهی با نام segment یاد می کنیم.

پس جدول یک segment است چون روی data file فضا می گیرد.

در سطح db چندین dbf خواهیم داشت آدرس این dbf ها را در کجا ذخیره می کنیم؟

در ساختار فیزیکی موجودیتی داریم به نام cf (control file) که (البته در حالت های مختلف چند cf داشته باشیم) وظیفه آن آدرس ذخیره سازی data file ها در داخل خودش است.

معمولاً control file از سمت instance از سمت data file مورد استفاده قرار نمی گیرد.

control file را می خواند آدرس db را پیدا می کند و حالا با db در ارتباط است.

فرض کنیم instance می خواهد یک رکورد را وارد یک data file کند در این حین برق قطع می شود از طرفی کار با data file ها بسیار کند است (به دلیل نوع ساختار ذخیره سازی که دارند)

حال چه باید کرد؟

Oracle یک قسمتی در ساختار فیزیکی دارد که به آن Redo log files گویند.

در اصل زمانی که با instance کار می کنیم اطلاعاتی که در سطح database وارد می کنیم اول داخل Redo وارد می شود بعد اگر نهایی شد در داخل data file ثبت می شود (یعنی زمان commit کردن آن به دیتابیس اضافه می شود)

اطلاعاتی که یک Redo log file به ما می دهد

یک data file قبل از transaction چه حالتی داشته است

Commit = به معنای انجام کامل عمل مورد نظر

در oracle سه R وجود دارد یعنی سه Redo log file وجود دارد که به صورت ترتیبی از آن استفاده می کند. فرض کنیم با insert می خواهیم رکوردی را به سطح data file ها اضافه کنیم. Insert اول به سطح Redo log file اول اضافه می شود. Commit که شد با insert دوم همینطور، ...، تا اینکه insert ششم انجام می شود. با 6 insert دیگر R پر می شود و اصطلاحاً به این mode گویند mode سوئیچ کردن Redo، oracle، Redo، switch را کرده و می رود روی Redo ی دوم.

و این عدد ۶ در واقع عدد ثابتی نیست ممکن است هر عددی باشد.

با پر شدن 2 Redo file روی redo3، switch می کند. دوباره redo3 پر می شود و سوئیچ می کند روی 1 Redo file و باعث از بین رفتن اطلاعات 1 Redo file می شود.

نکته: ما به عنوان admin می توانیم عمل switch کردن را انجام بدهیم. و علت های متفاوتی ممکن است باعث این عمل شود.

Redo خراب شده باشد.

احساس کنیم transaction کند دارد انجام می شود و البته اگر همه environmentها را چک کرده باشیم، مشکل از جای دیگری نباشد.

RDMBS = نرم افزاری که به کاربر امکان بالا آمدن instance را می دهد که آن نرم افزار oracle است.

Redo log R file یا اینکه database در مورد Rfile است یعنی چه؟

Database اوراکل این قابلیت را به ما می دهد که خودش به صورت خودکار از Redo فایل ها کپی بگیرد و نگهداری کند و به این فایل ها که نگهداری می کند اصطلاحاً Archive Redo Log می گویند و این کار به طور پیش فرض در بانک اطلاعاتی oracle انجام نمی شود و هنگامی که بانک اطلاعاتی oracle را نصب می کنیم موظفیم آن را به Rk، node، Rk ببریم تا اینکار انجام شود. زمانی که Redo، New می شود اینکار رخ می دهد در آن لحظه یک بار ایجاد می شود که به صورت مقطعی است که در آن لحظه دارد آن را archive می کند.

• اولویت با کپی کردن Redo نیست بلکه با انجام تراکنش است.

Automatic Storage: این قابلیت را اوراکل به admin می دهد که آیا فایل های db را توسط سیستم عامل مدیریت شود و یا توسط oracle؟

سرعت نوشتن در Redo file ها بسیار بسیار بالا است یعنی در هر شرایطی insert به Redo می رود ممکن است روی data file نرود ولی روی Redo file حتماً می رود.

Instance

را می توان به روش های مختلف بالا آورد یا خواباند.

روش هایی که instance با آن بالا می آیند: مثل روشن کردن ماشین دارای سه مرحله دارد.

حالت اول: اصطلاحاً در حالت Noumos است که در این حالت تنها چیزی که رخ می دهد این است که instance اطلاعات cl را فقط می خواند و می فهمد data file ها کجا هستند.

حالت دوم: حالت com است مرحله ای که cl file ها خوانده شده حالا data file ها را هم خوانده است.

حالت سوم: رسماً instance بالا می آید و آماده سرویس دهی شده است.

: Creating an oracle database

ساخت user و table و

ساخت user :

insert table.drop

constraint check

در محیط sql plus

create user amirsam identified By my java;

password username

User را باید dba کنیم و rule، connect را به آن assign کنیم با

اعطا کردن

grant dba to amir;

grant connect to amirsam;

حال می توانیم با دستور exit از سیستم خارج شده با کاربر amirsam وارد سیستم شویم.

امیرسام الان به صورت dba تعریف شده است پس می تواند یک user دیگر را تعریف کند به دو روش:

1. Create user amir identified by my java;

2. Create user amir identified externally;

حالت ۲ همان تعریف user به صورت OS است یعنی کسی که با windows، login کرده است می تواند به oracle با همان user و pass دسترسی پیدا کند.

حذف کاربر

drop user amir;

اگر این کاربر دارای segment باشد به صورت Automatic، drop نمی شود حتماً در انتهای دستور باید داشته باشیم:

Drop user amir cascade;

یعنی amir و متعلقاتش با هم حذف شوند.

ایجاد rule روی سطح database

نام rule

Create rule modiriat;

حال می خواهیم یک rule را به rule مدیریت assign کنیم.

Grant dba to modiria;

Rule dba را به rule modiriat اختصاص دادیم.

Grant connect to modiriat;

Create hasani identified by my java 123;

Grant modiriat to hasani;

مثل این است که dba connect را به مدیر connect کردیم.

گاهی لازم است که rule ی را بسازیم و در آن محدودیت هایی را ایجاد کنیم پس به صورت زیر داریم:

Create role mm;

grant SELECT, INSERT on amirsam.t1 to mm;

حال قابلیت select, insert در جدول t1 از کاربر amirsam را به رول mm دادیم.

وقتی rule dba را به کاربر اعطا کنیم by default, with admin option است ولی rule های دیگر را وقتی اعطا می کنیم with admin option نیست و خودمان به صورت دستی باید آن را به آن اعطا کنیم. مثلاً

grant mm to hassani;

حسنی نمی تواند این rule را به کاربر دیگری اعطا کند ولی بدانیم که rule dba به صورت اتوماتیک with admin option است.

امیرسام می تواند به حسنی rule mm را بدهد، حسنی در صورتی می تواند rule mm را به کسی بدهد که در آخر دستورش امیرسام بنویسد:

grant mm to hassani with admin option;

بازپس گیری نقش از کاربر (revoke)

با disconnect می توانیم از user خارج شویم.

گرفتن rule از کاربر حسنی

revoke dba from hassani;

revoke dba from mm;

grant select on amirsam,t1 to ali;

بهتر است که rule را مستقیماً به کاربر grant نمی کنند یعنی اول rule را به یک rule دیگه وصل کردیم سپس آن را به کاربر جدید اضافه کردیم. (کاملاً در ذیل توضیح می دهیم)

مثال:

create user ali identified by my java 123;

grant select on amirsam.t1 to ali; ❶

در اینجا rule را نساختیم بلکه مستقیماً عمل select را به کاربر علی assign کردیم.

grant connect to ali; ❷

حال اگر در کاربر amirsam دو جدول t1 و t2 را داشته باشیم، علی فقط به t1 دسترسی دارد.

و اگر با کاربر علی به بانک وصل شویم:

```
connect ali/my java 123;
```

```
select * from amirsam.t2; //
```

در اینجا دسترسی به t2 ندارد و error می گیرد.

```
select * from t1;
```

رکوردهای آن را نشان می دهد.

حال بهتر است به نام rule به ali rule بسازیم و سطح دسترسی های آن را معلوم کنیم و آن را به علی وصل کنیم به جای استفاده از دو خط ❶ و ❷

می نوشتیم:

```
grant select on amirsam.ti to ali rule;
```

```
grant connect to ali rule;
```

```
grant ali rule to ali;
```

دستورات اعطا کردن (grant) در اوراکل چه به user چه به rule یکی است.

دستورات بازپس گیری نقش هم چه به user چه به rule باز هم یکی است. (revoke)

حذف نقش:

با دستور drop انجام می شود.

```
drop role mm;
```

اگر user از آن rule استفاده کند به شما این اجازه را نمی دهد حتی اگر از cascade استفاده کنید!

اگر rule، assign شده باشد باید اول rule را revoke کنیم سپس نقش آن را حذف کنیم.

طریقه ساخت یک جدول:

```
create table amirsam. Table1 (id number, name varchar 2(20));
```

جدولی با دو فیلد id و نام به وجود آمد.

Number(10): یعنی تا ۱۰ رقم می توان در داخل این متغیر تعریف کنیم و عدد آخر در oracle llg، ۳۲ است یعنی نهایتاً عدد ۳۲ رقمی است.

قواعد default :

تعریف default value : یعنی یک مقدار پیش فرض در table داشته باشیم اگر هنگام insert آن فیلد را مورد توجه قرار ندهیم مقدار پیش فرض در داخل آن قرار می گیرد.

```
create table t1 (id number default 222, name vrchar2(20));
```

```
insert into amirsam.t5 (name) values ('akbar');
```

حال اگر داشته باشیم:

```
select * from t5;
```

ID	Name
222	akbar

خروجی:

اگر بخواهیم default کاراکتری تعریف کنیم تک کتیشن های آن هم باید بزنیم.

قواعد جامعیتی اطلاعات:

قاعده NOT Null :

```
create table amirsam.t6 (id number not null, name varchar2(20));
```

از این به بعد فیلد ID نمی تواند مقدار Null را بپذیرد.

قواعد جامعیتی (unique):

```
create table amirsam.t7 (id number unique, name varchar2(20));
```

از این به بعد ID نمی تواند مقدار تکراری بگیرد.

```
insert into t7 (id, name) values (1, 'hasani');
```

اجرا می شود

```
insert into t7 (id, name) values (1, 'hamid');//Error
```

چون ID را unique تعریف کرده ایم. اگر ۱ را تغییر بدهیم به ۲ درست می شود.

حال می خواهیم به constraint مان اسم بدهیم پس دستور را به شکل زیر اصلاح می کنیم:

```
create table amirsam.t7 (id number constraint H unique, name varchar);
```

حال اگر داده تکراری روی t7 وارد کنیم به error زیر بر می خوریم:

Amirsam.h

Amirsam مالک constraint ی به نام h است و error ی است که در قاعده جامعیتی پیش می آید.

Primary key

فیلد کلید اصلی دو خاصیت دارد:

۱. NOT Null

۲. Unique بودن

```
create tablet12 (id number primary key, name varchar2(20));
```

حال ID دیگر نمی تواند تکراری و Null باشد.

کلید خارجی (قواعد ارجاعی):

```
create table amirsam.t14 (id number primary key, Name varchar2(20), id-t13 references amirsam.t13);
```

در برخی مواقع می خواهیم رکورد اصلی که حذف شد رکوردهای وابسته هم حذف شود برای اینکار هنگام تعریف کلید خارجی ON DELET cascade را هم به تعریف اضافه می کنیم.

```
create table amirsam.t14 (id number primary key, Name varchar2(20), id-t13 references amirsam.t13 on DELET CASCADE);
```

حال اگر از جدول T13 رکوردی حذف شود که در جدول T14 رکورد وابسته ای به آن هم حذف می شود.

قاعده *check* در *database*

```
create table amirsam.t16 (id number, tel varchar2(20), constraint h check (id>2));
```

بهتر است *constraint* را در آخر دستور بنویسیم.

یادمان باشد زمانی که *validation* ها را سمت *db* پیاده سازی می کنیم، یک *connection* به *db* می زنیم یک *process* در آن انجام می دهیم، یک باری روی *db* ایجاد می کنیم، ممکن است یک وقفه زمانی برگشت خطا از سمت *db* به سمت *application* هم رخ دهد فقط برای اینکه در سمت *application* یک دستور ساده *if* و یا ... را استفاده نکنیم و این اصلاً کار مناسبی نیست.

پس بهتر است *check* را سمت *db* انجام ندهیم.

دستور *drop* برای حذف *table* به شکل زیر است:

```
drop table amirsam.t16;
```

دستور *describe*

به واسطه این دستور، وقتی که از ساختار این دستور مطلع نیستیم به واسطه این دستور ساختار جدول را می توانیم مشاهده نمائیم.

```
describe amirsam.t12;
```

Name	Null	Type
ID	NOT Null	Number
Name		Varchar2(20)

دستور *Data-Dictionary*

```
Describe dba-tab-cols;
```

در این جدول اطلاعات خیلی زیادی آمده است.

```
SELECT owner, table-name, column-name for dba-tab-cols where owner='amirsam';
```

یا `table-Name='T12';`

Structure ی برای جدول *T12* را می دهد.

dba-tab-cols یک table داخلی در خود oracle از پیش تعریف شده است که تمام ساختارهای table های دیگرمان در این table درج شده است.

اگر این دستور را بدون where بزنیم تمام جدول ها را برای ما می آورد البته با استفاده از کلید ctrl+c قطع جریان می کنیم.

درج داده

```
insert into amirsam.ti (id, name) values (1, 'hasani');
```

```
insert into t2(id, name) select id, name from t1;
```

از یک جدول فیلدهای بالا را در جدول t2 کپی می کند. (همه فیلدها را)

دستور update:

در دستوراتی مثل insert, update, delete بعد از اعمال دستور حتماً دستور commit را تایپ می کنیم وگرنه تغییرات در سطح db درج نخواهد شد.

چون sql plus, Auto commit آن false است. تا قبل از commit در Redo قرار داریم و بعد از commit می روی روی data file ها.

	Name	ID
	Hassani	1
T 1	Mahmodi	2
	Akbari	2
	hashemi	3

می بینیم که در جدول t1 در دو سطر یک ID آمده است و می خواهیم آن را اصلاح کنیم.

```
update t1 set id=4 where name='mahmodi';
```

```
commit;
```

از روی Redo به data file می رود.

```
delete from t1 where id=2;
```

```
select * from t1;
```

```
commit;
```

برای نشان دادن داده های جدول از select استفاده می شود.

```
select ID from t1;
```

where برای شرط گذاشتن است.

= مساوی

= < کوچکتر مساوی

> < نامساوی

= ! مخالف

= > بزرگتر مساوی

Oracle به حروف بزرگ و کوچک حساس نیست ولی نسبت به بزرگ و کوچک بودن داده ها حساس است مثلاً اگر در فیلد t1، نام را Maryam ذخیره کنیم با maryam فرق می کند.

برای رشته ها می توان از دستور like یعنی شبیه به استفاده کرد.

نام هایی که ۵ کاراکتری هستند:

`select * from T1 where Namr like '-----'`

چهار کاراکتری و آخرش i باشد.

Like '---i'

چهار کاراکتری آخرش i و اولش m باشد.

Like 'm---i'

Like 'm%i'

یعنی رکوردی که اولش m و آخرش i باشد بین این دو خالی باشد یا هر تعداد که خواست کاراکتر باشد مثلاً (mi, mai, mabi,)

برای پیدا کردن کلمه password که از ورودی از کاربر دریافت می شود از عملگر like استفاده نشود بلکه از علامت = استفاده شود چرا که اگر like بزاریم کاربر در ورودی % بذارد کاربر در سیستم login می شود.

(or به معنی یا) می توان در شرط ها از آن استفاده کرد (یکی از دو شرط مساوی بود اونو نشان بده)

(and به معنی و) (حتماً هر دو شرط برقرار باشد)

جدول dual : یک جدول مجازی است که هر کاری را می توان رو آن انجام داد.

select 2+2 from Dual; = select (2+2) jAM from Dual

$\frac{2+2}{4}$	خروجی	خروجی	$\frac{jAM}{4}$
-----------------	-------	-------	-----------------

پروسیجرها را روی این جدول اعمال می کنیم.

پرس و جو با استفاده از کلمه Distinct :

Distinct کمک می کند اگر رکورد تکراری در جدول باشد آن را یک بار نمایش می دهد.

select Distinct ID, name from t1;

عملگر +

به جای + می توان - x / را هم استفاده کرد.

select (ID+2) NID, Name from T1;

NID : اسم مجازی سطر است و (ID+2) جمع می کند

Concat : (II) علامت اتصال

برای اینکه رشته ای را به رشته ی در جدول مان اضافه کنیم:

select ID, 'mr.' II Name from t1;

select ID, ('mr.' II Name) realname from t1

کلمه mr. را به ابتدای name اضافه می کند.

Null (is not null هست) و Null (is null عملگر نیست):

insert into t1 (id, name) values (null, 'sami');

select * from t1 where id is Null

ID	Name
	sami

عملگر in: یعنی آنهایی که در یک بازه ای هستند.

select * from t1 Name in ('hasani', 'mahmodi');

اسم آنها یا mahmodi یا hasani است.

NOT IN هم عکس آن است.

برای اعداد هم می توان از between استفاده کرد.

```
select * from t1 where id between 1 and 4
```

بازه آنها از ۱ تا ۴ است را بر می گرداند.

Union: اجتماع: اگر رکوردی در داخل جدول t2 داشته باشیم

```
select * from t2;
```

```
union
```

```
select Name from t1;
```

تکراری ها فقط یک بار می آیند.

اجتماع دو جدول را نشان می دهد و رکوردهای تکراری را هم نشان می دهد: union All

```
select * from t1;
```

```
union All
```

```
select * from t2;
```

همه رکوردها + تکراری ها

Intersect : اشتراک

```
select * from t1;
```

Intersect

```
select * from t2;
```

فقط مشترک های جدول را نشان می دهد.

MI-NUS : از هم کم می کند.

```
select * from t1;
```


MI-NUS

```
select * from t2;
```

همه رکوردها = مشترک ها را می آورد.

توابع:

تاریخ جاری سیستم:

```
select Date from Dual;
```

کاربر را به سیستم برمی گرداند:

```
select user from Dual;
```

تاریخ فارسی سیستم را بر می گرداند:

حالت اول:

```
select to-char (system, 'yyyymmdd', 'nls-calender=persian') from Dual;
```

حالت دوم:

```
ALTER SESSION SET NLS-CALENDAR='persian'
```

کل ارتباط را به صورت فارسی می کند:

```
select system from dual;
```

اگر حالا از سیستم disconnect شیم با user سیستم connect بشویم دوباره همه چیز به حالت اول برمی گردد.

Alert session یعنی در همین جلسه کاری

تابع CHR

```
select chr (id) from t1;
```

CONCAT (اتصال II)

```
select CONCAT ('mr.' , name) from t1;
```

تابع soundex

تلفظ یک سری از رشته ها را بر می گرداند.

```
select * from t1 where soundex (name) = soundex ('mehmody');
```

ID	Name
4	Mahmodi

تابع lower :

کوچک کننده حروف

```
select lower (name) from t1;
```

حروف را بزرگ نشان می دهد:

```
select upper (name) from t1;
```

حرف اول را بزرگ باقی را کوچک نشان می دهد:

```
select initchar (name) from t1;
```

: lpad

```
select lpad (name, 20, '*') from t1;
```

۲۰ تا * قبل ا هر اسمی قرار می دهد و آن را نشان می دهد می توانیم به جای ستاره از یک رشته هم استفاده کنیم.

: Rpad

از سمت راست به هر تعداد رشته اضافه می کند.

```
Rpad (name, 20, '*')
```

تابع ltrim :

Space را از سمت چپ از بین می برد

Rtrim هم space های راست را حذف می کند.

Trime : از سمت راست و چپ space ها را حذف می کند.

: Replace

```
select replace (name, 'mi', 'gy') from t1;
```

به جای mi حروف gy را بگذار.

توجه داشته باشیم این تغییرات فقط در سطح select است و در جدول اعمال نمی شود.

subSTR

از کاراکتر اول به بعد ۳ تا را نشان می دهد:

```
Select substr (name, 1, 3) from t1;
```

از کاراکتر سوم یکی را نشان می دهد:

```
Select substr (name, 3, 1) from t1;
```

INSTR : همان index of است.

از کاراکتر اول شروع کن و جایگاه اولین کاراکتر h را به عدد به من نشان بده.

```
Select INSTR (Name, 'h', 1,1) from t1;
```

از کاراکتر اول شروع کن و جایگاه دومین کاراکتر h را به عدد به من نشان بده.

```
Select INSTR (Name, 'h', 1,2) from t1;
```

Length : تعداد کاراکترها را به ما بر می گرداند.

```
select length (name) from t1;
```

توابع ریاضی:

➤ Ceil :

عدد را به عددی بزرگ تر از خودش round می کند.

```
Select ceil (id) from t1;
```

-5.65=-5

0/33=1

➤ Floor : یک عدد را به عدد کوچک تر round می کند.

➤ Round : تا 49. را به عدد کوچک تر و 50. را به عدد بزرگ تر round می کند.

➤ Power : به توان می رساند.

Select power (id,2) from t2;

➤ Sqrt : جذر را بر می گرداند.

➤ ABS : قدر مطلق اعداد را بر می گرداند عدد منفی را مثبت می کند.

Select abs (id) from t1;

➤ Sign : اعداد بزرگ تر از صفر را به یک عدد کوچکتر از یک را به منفی یک و به ازای عدد مساوی صفر صفر برمی گرداند.

Select sign (id) from t1;

➤ MOD : باقی مانده تقسیم را بر می گرداند.

Select MOD (id, 2) from t1;

توابع sin و cos و tan و cot که برحسب رادیان بر می گرداند و با ضرب در 0.01745329251994 کنیم تا نتیجه به صورت درجه برگردد.

➤ Exp : یک عدد ثابت $e=2/718282$ است عدد e به توان ID می رسد.

Select EXP (ID) from t1;

➤ Log : log طبیعی عدد را بر می گرداند.

(با دستور Roll back دستوری که نوشتیم مثلاً delete ی که می خواهیم برگردانیم به حالت قبل)

توابع تاریخ و زمان:

Create table my date (id, number, START-Date Date, End-Date Date);

Insert into my date (ID, start-date, End-date) values (ID, sys date, sys date+2);

در اوراگل وقتی date را به اضافه ۲ می کنیم می فهمد که آن را باید بر مبنای تاریخ انجام دهد نه ریاضی.

و جمع را بر مبنای روز انجام می دهد مثلاً ۳۶۵ به معنای یک سال است.

همچنین این کار برای – هم امکان پذیر است و از تاریخ کم می کند.

تابع MONTHS-between

Select months-between (start-date, End-date) from my Date;

تعداد ماه های بین دو تاریخ را بر می گرداند.

Select ADD-months (start-Date, 2) from my date;

دو ماه به ماه اضافه می کند.

تابع last-Day

آخرین روز ماه را بر می گرداند.

تابع Next-day یک روز ما را به جلوتر می رود.

Select Next-day ('1 FEB08' , 'Tuesday');

یعنی تاریخ اولین Tuesday بعد از فوریه ۲۰۰۸ را به من برگردان.

Select Next-day ('sus-Date, "FRIDAY") from Dual;

تاریخ اولین جمعه نسبت به تاریخ امروز سیستم را بر می گرداند.

جستجوی انعطاف پذیر:

جدولی به نام person مفروض است.

Person

ID, Name

Select * from person where Name='& your Name';

استفاده از پارامتر & در ابتدای your Name به این معناست که پارامتر از ورودی دریافت شود.

ادغام دو جدول

: ضرب دکارتی:

دو جدول information و person مفروض است:

Select * from person, Information;

هر رکورد از جدول اول در رکوردهای جدول دوم ضرب شده است.

Select * from person, Information where person. ID=Information.ID-person

حال حالت اصلاح شده را می گیریم (بدون ضرب دکارتی)

Select Name, Name, ID from person, Information where

این کار قابل انجام نیست چون بین Name و ID ها فرقی نداریم باید نام جدول قبل از همه آنها بیاید.

Person.ID

Person. Name

Insert into person (Name, ID) values ('Sami', 50);

Insert into person (Name, ID) values ('Akbar', 60);

Select * person

داده های table را نشان می دهد ولی اگر در انتهای insert ننویسیم commit; داده های table ثبت نهایی نمی شود.

تغییر داده با استفاده از دستور update :

Update person set Name='sami' where ID=1;

برای کپی کردن تمام رکوردهای یک جدول در جدول دیگر از دستور دیگری استفاده می کنیم.

جدول Hashem مفروض است:

INSERT INTO HASHEM (ID, NAME) (Select ID, Name. from person);

حتی می توانستیم این insert را با قرار دادن where محدود کنیم.

حذف کردن رکورد

DELETE from hashem where ID=2;

ایجاد جدول:

Create table amirsam (ID Number, Name varchar2(20);

ایجاد جدول با مقدارهای پیش فرض:

Create table amirsam (ID Number Default '222', Name varchar2(20) Default 'Ali');

زمانی default مناسب است که هنگام insert کردن به table اسم اون فیلد مقدار نداشته باشد و اصلاً در دستور insert قید نگردد.

ایجاد محدودیت در جدول:

Create table hasani (Name varchar2(20) NOT Null, ID Number);

حال نمی توان فیلد نام را Null رها کنیم.

ایجاد یک جدول یا فیلد unique :

Create table b (ID varchar2(20) constraint HHH unique, ID Number);

پس ID مقدارهای تکراری را نمی پذیرد.

تعریف کلید اصلی:

Create table b (ID Number primary key, name varchar2(20));

ساخت کلید خارجی:

فرض کنیم یک جدول به نام (ID, name) information داریم حال

Person (ID, name, ID-information)

Create table person (ID Number primary key, Name varchar2(20), ID information constraint HEE references information (ID));

- **HEE**: اسم constraint

- **References**: متصل کن

- **Information ID**: به جدول information

- اگر به فیلد خاصی بخواهیم وصل کنیم نام آن را به صورت (ID) کنار نام جدول می نویسیم.
- اگر نام فیلد خاصی را مطرح نکنیم Auto آن را به pk آن جدول وصل می کند.
- کلید خارجی فقط و فقط برای جامعیت داده در زمان درج اطلاعات درجه اول مناسب است.
- زمانی pk را تعریف می کنیم که بخواهیم از ورود اطلاعات نامتعارف جلوگیری کنیم.

Insert into person (ID, Name, ID-information) vlues (1, amirsam, 31);

می دهد. Error

Insert into information (ID, Address) values (1, Tehran);

Insert into person (ID, Name, ID-information) values (1, amir, 1);

: Sequence

در db های دیگر ما شمارنده داریم اما در oracle آن را نداریم و باید خودمان آن را به وجود بیاوریم.

Create sequence mySEQ START with 1 increment by 1;

ایجاد view برای جدول:

View چیست؟ یک نمایی از یک جدول است و مثل یک filter روی جدول عمل می کند.

اگر رکوردی در جدول اصلی حذف شود در view هم حذف می شود.

با view می توانیم محدودیت در نمایش اطلاعات ایجاد کنیم.

آیا در view می توان delete و insert و update می توان انجام داد؟ بله و اینکار بر روی جدول اصلی تأثیرگذار خواهد بود.

دو نوع روش برای نگهداری اطلاعات داریم:

۱. Backup : فقط data را نگهداری می کند.

۲. Export : data و export را نگهداری می کند. در این روش هم دو دسته داریم: کلاسیک و data form.

در روش کلاسیک سرعت انتقال داده (در export گیری) بسیار کند است و یا اگر بخواهیم تند باشد منابع سیستمی درگیر می شود ممکن است oracle دیگر export نگیرد در روش دیگر منابع سیستمی dynamic تغییر پیدا می کنند.

rule → Revoke بازپس گیری نقش

Save point نقطه ای را مشخص می کند.

Save point x: اگر آن را تایپ کنیم با زدن enter اجرای دستورات به نقطه x برمی گردد.

اگر یک user همانم user که قبلاً تعریف کرده باشیم دوباره ایجاد کنیم دیگر لازم به تعریف password آن نیست.

Create table T999 (ID Number) STORAGE (INITIAL) 100M NEXT 20M MAX EXTENTS 200);

برای سنت table space و assign کردن آن به کاربر وارد محیط Inter price Manager می شویم.

فصل پنجم

j SQL

Java- Core

پس از ایجاد یک پروژه روی پوشه SRC راست کلیک کرده و new java class را انتخاب می کنیم.

و Tehran. amirsam را تایپ می کنیم به این شکل است که package به نام Tehran و کلاسی به نام amirsam ایجاد می شود.

```
Public static void main (string ja){  
  
System.out.println ("hello");  
  
{
```

II : علامتی برای نوشتن توضیحات در برنامه

انواع متغیرها در جاوا:

Long : برای نگهداری اعداد بسیار بزرگ

Double : اعداد اعشاری بزرگ

Int : برای نگهداری اعداد صحیح

Boolean : true و false

Float : برای نگهداری اعداد اعشاری

String : برای نگهداری رشته.

انواع عملگرهای محاسباتی:

*, / , تقسیم، جمع + ، % باقی مانده، - منهای، + اضافه کردن یک واحد، - - کم کردن یک واحد

ساختار if در جاوا:

```
if (Num == 2)
{
    // ...
}
else {
    // ...
}
```

ساختار دستور *switch*

```
Switch ( ) {
    Case1:-----
        break;
    Case2:-----
        break;
    :
    :
    Default:
        -----
        -----
}
```

حلقه ها:

اگر شرط درست برد داخل حلقه اجرا شود:

```
While ( ) {
    // ...
}

for (int i=0; i<= 10; i++ ) { }
```

آرایه:

یک آرایه ای از a با ۸ تا خانه ساخته شد.

```
String [ ] a=new string [8];
```

```
A[0]="Amir";
```

```
A[3]="hasan";
```

مفهوم کلاس:

Amirsam یک کلاس است با یک متد به نام متد main است.

```
Public class Amirsam {
```

```
Public Static void main (String [ ] jargs){
```

```
Amirsam. ghaza khordan
```

```
}
```

```
Public static void ghaza khordan {
```

```
System. println ("yam yam");
```

```
}
```

متد می تواند دارای ورودی و خروجی باشد.

```
Public static void sum (inti, intb){
```

```
}
```

متدی است با دو ورودی ولی بدون خروجی.

برای قرار دادن خروجی برای متد، به جای void از نوع خروجی مورد نظر را نام ببریم مثلاً string و int و

کلاس می تواند دارای خصوصیت باشد مثلاً s یک خصوصیت است.

Public static Strings;

در جاوا دو حالت خصوصیت داریم: خصوصیت static و خصوصیت غیر static

فرض کنیم خصوصیت های زیر را در کلاس مان تعریف کرده باشیم:

```
Public class Amir{

    Public static String s;

    Public String s2;

    Public static void main (string [ ] args){

    }
```

در این حالت در داخل متد main به متغیر غیر static دسترسی نداریم. چرا که متد main از نوع static است و متدهای static تنها به خصوصیات و متدهای static به صورت عادی دسترسی دارند.

در صورتی که از روی کلاس object بسازیم به خصوصیات غیر static ی آن هم می توانیم دسترسی داشته باشیم.

```
Amirsam javad = new Amirsam ( );
```

```
Javad.S2= 1;
```

```
Javad.S= 2;
```

وقتی متغیری static باشد به ازای تمام instance هایی که از روی کلاس ساخته می شود فقط یک متغیر static داریم پس همیشه در آن متغیر آخرین مقدارش قرار می گیرد و همه instance ها به آن دسترسی پیدا می کنند.

ولی در غیر استاتیک ها هر متغیر در هر instance مقدار خاص خود را داراست.

پس تغییر متغیر static در همه instance ها تأثیر می گذارد. و در غیر static هر شی برای خودش یک خصوصیت خاص خودش دارد.

متدی به نام constructor داریم: متدی است که خروجی ندارد (حتی void هم نیست) در زمان ساخت شی متد static، call م شود.

```
Public class Amirsam {

    Public Amirsam ( ) {

    Public static void main (string [ ] args) {

        Amirsam. Javad= new Amirsam ( );
```

در حال حاضر ما فقط از روی کلاس یک شی ساخته ایم.

این باعث می شود خود به خود متد constructor فراخوانی و اجرا شود.

پس متد سازنده هم نام کلاس است و خروجی ندارد.

ارث بری: فرض کنیم کلاس دیگری به نام کلاس sohile ایجاد کنیم.

```
Public class soheil{

    Public void a ( )

    {

    }

    Public void b ( ) {

    }

}
```

حال اگر امیرسام از کلاس سهیل ارث ببرد به برخی از توابع و خصوصیات آن دسترسی پیدا می کند.

```
Public class Amirsam extends soheil{

}
```

چون از سهیل ارث برده است دقیقاً به متدی آن دسترسی پیدا می کند.

در جاوا هر کلاس می تواند دارای یک پدر باشد نه بیشتر مگر پدران از جنس کلاس نباشند.

: Interface

```
Public class cat implements animals, Entity{
```

تمام متدهای موجود در interface ها را باید در این جا پیاده سازی کنیم.

```
}
```

یادمان باشد که متد در interface، body ندارد.

```
Public interface Animal{
```

```
Public void Tanafos ( );
```

```
Public void Davidan ( ); }
```

برای ارث بری از interface از کلاس implements استفاده می شود.

```
Public interface Entity {
```

```
Public void Zende boodan ( );
```

```
Public void ghaza khordan ( );
```

دسترسی در تمام پروژه

```
Public class Reza {
```

```
Public void a ( ) {
```

```
System.out.println ("a");
```

```
}
```

دسترسی فقط در همان کلاس

```
Private void b( ) {
```

```
System.out. println ("b"); }
```

package : دسترسی در سطح همان

```
Protected void c( ) {
```

```
System.out. println ( );
```

```
}
```

کلاس انتزاعی:

قانون اول: کلاس abstract از روی آن شی ایجاد نمی شود. بلکه فقط به درد پدر شدن می خورد.

و می توان از آن ارث بری کرد.

کلاس abstract می تواند دارای متد abstract باشد و متدی که abstract باشد فرزند موظف است پیاده سازی آن را بر عهده گیرد.

Try { } catch { }

گاهی اوقات در برنامه خطایی رخ می دهد که به آن خطای استثنائی گویند چون شما در ایجاد آن نقشی نداشته اید شما موظف به مدیریت آن خطا هستید و جاوا به شما این اجازه را نمی دهد که آن را مدیریت ننمائید.

Class amir () {

try {

} catch (Exception e){

}

}

Public static void main (string { } args) {throws Exception {

c.c ();

}

Public void throws Exception

{

System.out. println (1μ);

}

زمانی که برنامه نویس در کدزنی تنبلی به خرج می دهد و Error ها را مدیریت نمی کنند از این جمله استفاده می کنند. در اینجا exception، handle نشده است چرا که خطا را به main برگردانیدیم main هم آن را به OS برمی گرداند.

نکته: برای تبدیل string به integer داریم:

```
String name="12";
```

```
Int i=Integer.parseInt (name);
```

```
Float F=float.parseFloat (name);
```

```
Int x= 12;
```

```
String h= x + " ";
```

هر data type ی به اضافه + " " خروجی اش string می شود.

```
Array list a= new Array list ( );
```

```
a.add ("Amirsam");
```

```
a.add ("bahador");
```

به محیط pl sql Developer می رویم:

در گوشه سمت چپ (All object) java source را کلیک راست می کنیم. New بزنی اسمی را وارد می کنیم. مثال:

Create or replace and compile java source named rr as

```
Public class rr
```

```
{
```

```
Public static string entry ( )
```

```
{
```

```
return "hello";
```

```
}
```

```
}
```

با زدن دکمه F8 کامپایل انجام می شود و این کد در اوراکل ذخیره می شود.

روی function راست کلیک New کرده

Create OR replace function salam return varchar2 as language java name

‘rr. Entry () return java. Lang. string’;

در اصل ما برای دستور rr. Entry یک نام مجازی به نام salam درست می کنیم. خروجی متد entry از نوع string بوده و متد salam از نوع varchar2 است.

حال در cmd به sql plus می رویم:

Select salam () from doual;

Salam ()	
hello	خروجی

فرض کنیم در داخل java source یک کلاس دیگر به نام Hossein بسازیم.

حال در کلاس hasan از آن یک object ایجاد می کنیم.

```
Public class hasan{
Public static string entry ( ) {
    Hossein h= new Hossein ( );
    return 2+2+h.f ( );}
```

```
Class hossein {
Public string f( )
{
    return “f .....f”;
}
```

حال در sql plus

{

Select salam () from doual;

SALAM ()

4 h F F F -----

مثال:

Create or replace integer haha

Befor insert on Amirsam.T1

For each row

Declare

Begin I varchar2(20);

I: = salam ()

Dbms- output. Put- line (I)

End hahal

همین جا sql window را باز می کنیم و یک insert می زنیم.

Insert into Amirsam.T1 (ID) values (1);

با رفتن به output خروجی را می بینیم.

java native : Sql j

نوع دوم JDBC است.

در style دوم JDBC کار به این شکل دنبال می شود.

نصب

بر روی next کلیک می کنیم دکمه yes را می زنیم بر روی دکمه create and configure a database کلیک می کنیم در اینجا می گوئیم که برای ما یک مخزن هم ایجاد کند.

Install database software only (فقط نرم افزار دیتابیس را نصب می کند بدون مخزن)

Upgrade an existing database (از database دیگری upgrade اش کن)

Next را می زنیم. بر روی server class کلیک می کنیم. Next را می زنیم. در تب دیگر دو گزینه داریم single instance و Real application chaser (برای multi instance ها و rack است) ما که تگ است گزینه اول یعنی single instance را می زنیم. Next را می زنیم گزینه Advanced install را انتخاب می کنیم. در تب بعدی زبان را English انتخاب می کنیم. در تب بعدی نسخه edition Enter price که کامل ترین نسخه است را انتخاب می کنیم next را می زنیم از ما مسیر نصب db را می پرسد در text box اول در واقع Base root را می بینیم یعنی oracle را در آن مسیر نصب می کند و در text box پائینی نرم افزار اوراکل را نصب می کند.

فرض می کنیم داریم server, config می کنیم پس در یک drive یک پوشه به نام oracle RDBMS می سازیم و نصب را در این پوشه انجام می دهیم پس آدرس Base root را مساوی با آدرس این فولدر oracle RDBMS قرار می دهیم. D\Root\oracle RDBMS حال next را می زنیم.

اگر روی database زیاد Query می گیرید و به عنوان یک مخزن داده زیاد استفاده می شود روی گزینه دوم کلیک کنید یعنی گزینه Data were housing، اگر database معمولی است و به عنوان مخزن داده استفاده می شود و روی آن transaction زیادی دارید روی گزینه general purpose کلیک کنید.

در این تب اسم اولین مخزن را از ما می پرسد (orcl)

پیشنهاد می شود نام آن را تغییر ندهیم بر روی next کلیک می کنیم.

در این تب می خواهد فضای PGA را config کند. همچنین می بینیم مقدار Allocatmen که اختصاص داده شده است، 1GB است که اتوماتیک بین SQL و PGA تقسیم می شود.

SID در اصل service ID ما است SID و Global همیشه یک مقدار دارند.

Next را می زنیم.

فضای (SGA) system global area فضایی که سه عنصر مهم در آن قرار دارد.

۱. Private global area (PGA)
۲. share buffer
۳. data pool buffer
۴. java buffer

دستورات sql و plsql در محفظه share buffer قرار می گیرند مستقر در فضای SGA و SGA مستقر در RAM است.

Data pool buffer : زمانی که یک select می گیریم کل رکوردها را از روی یک data file روی یک باقری کش می کند و از این به بعد Query ها از آن cache گرفته می شود اسم این فضا Data pool buffer است.

Java pool buffer : زمانی که sql کار می کنیم object هایی که در آن تولید می کنیم در java pool buffer نگه داری می شوند. تمام اینها مستقر در فضای SJA هستند.

زمانی که select می زنیم از Data pool buffer داریم select می زنیم حال از روی همین data pool buffer یک where می زنیم پردازشات where روی فضای PGA از اوراکل خواهد بود.

در oracle IIg به این شکل عمل می کند که Auto Allocate می کند یعنی یک مقداری را در نظر می گیریم مثلاً ۱۰۲۴، هنگامی که فضای حافظه کم باشد فضای SGA را زیاد می کند.

Next را می زنیم (نکته: اگر سرور real server دیتابیس ۴۰٪ از RAM را به ۷۰٪ تبدیل کنیم ۳۰٪ برای مصرف OS است) next را می زنیم.

در این تب اوراکل به ما این قابلیت را می دهد که اگر رخدادی در سطح اوراکل رخ داد به Email, admin بزند.

Next را می زنیم، ما گفتیم orcl شامل دو قسمت است یک قسمت instance است و قسمت دیگر قسمت فیزیکی است (یک قسمت فیزیکی و یک قسمت منطقی) است.

Data file ها:

همان واحدهای فیزیکی database هستند. در قسمت special database آدرس data file ها را از ما می پرسد.

: Automatic Storage men

اوراکل این قابلیت را دارد که داده ها را روی دیسک های فرمت نشده بخواباند و تکنولوژی به نام ASM دارد. اگر طرف حساب ما یک دیسک خاصی بود روی این گزینه کلیک می کنیم اما به صورت پیش فرض ما همیشه روی file system کلیک می کنیم. Next را می زنیم.

در این تب قابلیت back up گیری بر طبق یک برنامه خاص به صورت دوره ای را اهدا می کند. و چون ما خودمان در آینده به صورت دستی انجام می دهیم گزینه don't enable کلیک می کنیم. Next را می زنیم. در قسمت بعد pass را از ما می خواهد.

(عدم امن بودن pass را هم به ما ممکن است هشدار دهد که برای ما مهم نیست)

دکمه finish را می زنیم. نصب پایان یفت.

در مسیر start روی oracle-oraDbilg-hom1 ← Data Base Control-ord کلیک می کنیم و محیط تحت وب برای مدیریت اوراکل را می بینیم
login ← sys و user

در قسمت performance وضعیت RAM را نشان می دهد. با قسمت server بسیار سر و کار داریم مثلاً در قسمت storage روی control file می توانیم کلیک کرده و مدیریت داشته باشیم همچنین روی table space ها مدیریت داشته باشیم.

مثلاً با کلیک بر روی create table space به یک صفحه جدید می رویم نام آن را وارد می کنیم در قسمت storage آن data file را Add می کنیم
Ok ← continue

در قسمت security می توانیم user و role و profile و ... را همین جا ایجاد کنیم.

در قسمت schema هم می توانیم table بسازیم.

ساخت repository :

Start → oracle-DraDbllg-home1 → configuration and migration tools → Database Configuration Assistant → Next

Create dabs ← گزینه Next ← general data را می زنیم ← SID را وارد می کنیم.

همان تنظیمات ارسال Email و Back up و را از ما می پرسد. User و pass را وارد می کنیم.

System storage را file system را انتخاب می کنیم و ← در ادامه مخزن جدید ایجاد می شود (مخزن در p).

زمانی که در مخزن هستیم وقتی به sql plus وصل می شویم دیگر نمی توانیم بگوییم user: system و pass:

حتماً باید بگوئیم system word (اسم مخزن) و orcl همیشه به instance ل اشاره می کند.

۲۲ : ۳۳

intell idea در JDB Example

برای oracle کارها بهتر است که از همان jdk ی که در oracle هست استفاده کنند چرا که می خواهیم jar file بسازیم بهتر است ورژن jdk با jdk در اوراکل set باشد.

```
Import java. Sql. *;
```

```
{
```

```
Public class Amirsam{
```

```
Public statement statement;
```

```
Private connection connection;
```

```
Public Amirsam ( ) throws Exception {
```

```
Class. For Name ("oracle. Jdbc. Driver. Oracle Driver");
```

```
Connection = Driver Manager. Get connection ("jdbc: oracle. Thin.c)
```

```
Local host: 1521: orcl, "syswsysdba", "my java123");
```

```
Statement= connection. Create statement ( )
```

```
{
    Public void insert (string name) throws Exception
    Statement. Execute update ("insert into T1 (ID, Name) values
    (1, ' "name" "sami" ');
    Public void close ( ) throws Exception {
        Statement. Close;
    Connection. Close;
    {
    Public static void main (string [ ] args) throws Exception {
        Amirsam amir=new Amirsam ( );
        amir. Insert ("sam");
        amir. Close ( );
    }
```

sqlj به معنای function نویسی به واسطه زبان برنامه نویسی جاوا در سطح دیتابیس:

ورود به محیط pl, sql Develop

در تب All object روی java source کلیک راست کرده New را می زنیم template window را باز می کنیم یک کلاس می سازیم نام آن را وارد می کنیم class را خالی می گذاریم ok را می زنیم.

Create or replace and compile java source named **rr** ahmad as public class **rr** ahmad

یک جاوا سورس را کامپایل و ایجاد و replace کن با نام Ahmad .

متدی با نام entry دارد که کلمه hello را return می کند.

```
{
    Public static string entry ( ) {
        Return "hello";
    }
```

{

F_n را نگه می داریم تا کلاس احمد را در سطح دیتابیس ایجاد کند در مرحله بعد متد entry را به عنوان یک function برایش تعریف می کنیم.

پس در قسمت function کلیک راست کرده گزینه New را می زنیم نام را وارد می کنیم ok را می زنیم.

Create OR REPLACE **help word** entry RETURN VARCHAR2 AS LANGUAGE JAVA NAME '**it** AMHAD. Entry ()
return java. Lang. String' ;

String : نوع خروجی

$F_8 + F_n$ را می زنیم.

وارد محیط cmd (sql pluss می شویم)

Select entry () from dual;

Entry ()
خروجی -----
Hello ;

حال اگر متد entry ورودی داشته باشد مثلاً string name چه کنیم؟

Public static string entry (string name) {

return "hello + name";

{

حال که متد دارای امضا است باید پارامترهایش را معلوم کنیم حال به قسمت function می رویم.

Create
Lang

Entry (name varchar2) Return varchar2 AS
Ahmad. Entry (java. string) return java. lang. string
در محیط sql plus

Select entry <'sami'> from dual;

Entry ('sami')

Hello sami

همچنین مدل دیگری از کد زدن داریم که توصیه نمی شود.

Public static string entry (string name)

```
{
    Try {
        Int id=0;

        Int x=900;

        Sql { SELECT max (id) into: id from T2 where id <:x };

        return id+ “!!!”;

        {catch (Exception e )

            {return “false” +e. get message ( ): }

        }
    }
}
```

#sql یعنی این قسمت از کد، کد sql است نه java

Max (id) را بگیر و توی id که در java هست بخوابان.

نوع دوم sqlj :

در این روش همان متد entry را داریم ولی body آن را clear می کنیم.

Public static string entry (string name)

```
{
    try {
        connection conn= Driver Manager. get Connection (“jdbc: default: connection:”);

        یعنی از بیرون به اوراکل وصل نشو، تو در خود اوراکلی و هیچ user و pass لازم نیست و همچنین نیاز به .... class for نیست.

        Statement stmt= conn. Create statement ( );

        Result Set r= stmt. Execute Query (“select * from T2”);

        String name1=” “;

        While (r. next ( ) )
```

```
{
```

در اوراکل ۱۶ روش connection به اوراکل داریم این روش internal connect است.

```
    Name1= r. get string ("name");
```

```
}
```

```
    Stmt. Close;
```

```
    return name1;
```

```
{catch (Exception e)
```

```
    return "khata" + e. get Message ( );
```

```
}
```

ساخت *jar file* در *oracle*

در محیط intelli jead برای پروژه مان می خواهیم jar file بسازیم به قسمت project structure می رویم.

(در جاوا jar file ها هم حکم فایل اجرایی را دارند هم فایل کتابخانه ای)

Artifacts → + → jar → from dependency module → ok

تیک build on make را می زنیم ok را می زنیم یک بار پروژه را Rebuild می کنیم و به آدرس مورد نظر رجوع می کنیم.

حال می خواهیم این jar file را به اوراکل بدهیم:

به محیط cmd می رویم (cmd نه sql plus)

Load java d:\ss. Jar

در محیط pl/sql با user : sys وارد می شویم

→ Java source → ماژول جدید →

در این جا می توانیم از متدهایی که در jar فایل آورده بودیم استفاده کنیم.

RMan 1

در محیط RMan پس از back up گیری محلی برای ذخیره سازی back up وجود دارد. چگونه در اوراکل چگونه محل default برای ذخیره سازی را بیابیم؟ برای این منظور از static data dictionary استفاده می شود که یک سری view هایی هستند در داخل اوراکل که می توانیم آنها را با یک سری دستورات تنظیم کنیم. الان می خواهیم یک command بزنیم که به ما نشان بدهد data file هایمان کجاست برای این کار از دستور sql استفاده می کنیم. ابتدا به محیط sql plus متصل می شویم.

Sql plus / as sys dba

اگر بخواهیم با user root به اوراکل وصل شویم از این روش استفاده می شود یعنی در واقع از دسترسی سیستم عامل برای وصل شدن به اوراکل استفاده می کنیم.

Name	Null?	Type

Describe v\$ recovery- file- dest;

Select name v\$ recovery- file- dest;

این دستور محل ذخیره سازی back up را به ما نشان می دهد.

D\ Root\ Oracle RDBMS\ flash- recovery- area خروجی :

یعنی ما اگر الان دستور RMan بک آپ، را صادر کنیم در همین محل اجرا می شود.

پس در همان مسیر در فولدر oracle RDBMS یک فولدر درست می کنیم و اسم آن را back up می گذاریم.

Alter system set db- recovery- file- dest= "D:\Root\oracle RDBMS\ back up";

دوباره اگر با دستور select name from می بینیم که مسیر را به فرم زیر تبدیل کرده است.

D:\Root\oracle RDBMS\back up

کنترل فایل ها را از کجا می توانیم بخوانیم؟

Name	Null?	Type

Describe v\$ control file;

Select name from v\$ control file;

RMan2

همچنین برای پیدا کردن محل data file ها به همین طریق عمل می کنیم.

Select name from v\$ data file;

فیزیکال از سه واحد (کنترل فایل ها، دیتا فایل ها؛ ری دو لاگ ها) تشکیل شده است.

Redo log ها به صورت ترتیبی مورد استفاده قرار می گیرند. ترانزکشن که انجام می شود اول روی Redo و بعد اعمال می شود روی data file ها.

- کنترل فایل در اصل مسیر این data file ها را نگه داری می کند. که کدام data file کجا هست؟!
- یکی از ایرادات وحشتناک خراب شدن control file است اگر data file خراب شود مشکل آن چنانی پیدا نمی شود و می توانیم recovery کنیم ولی control file که خراب شود اگر back up نگرفته باشیم همه چی از دست رفته است.
- پس هنگامی که insert انجام می دهیم اول روی Redo بعد روی data file اعمال می شود و همچنین به شرطی که data base در مد archive (آرشیو) باشد می توانیم از روی Redo ها یک کپی بگیریم.
- کاری که می خواهیم انجام دهیم- ایجاد table - گرفتن back up - بعد جدول را حذف می کنیم- back up را restore می کنیم ببینیم جدولمان برمی گردد یا نه.
- (بک آپ گیری با کنترل فایل)

T 1	ID
خروجی	1
	2
	3

Create table t1 (id number);

Insert into t1 values (3);

Commit;

Start up این پارامتر فایل را به نام ora.init را می خواند (فقط می خواند کاری با data file ندارد)

Start up سه مرحله دارد:

۱. Start up no mount

۲. Start up mount

۳. Start up open

در ۱ کنترل فایل ها را می خواند ۲ در این حالت data file ها را می خواند ۳ دسترسی را با کاربرها ایجاد می کند.

اگر Start up خالی را بزنیم هر سه مرحله را بدون وقفه پشت سر هم انجام می دهد.

Start up mount کنترل فایل هام رو بخون، بازش کن، data file هامو بخون اما بازش نکن.

اگر data base ما بالا بود آن را خاموش می کنیم با دستور ← shut down immediate;

انواع shut down در data base :

Shut down ← زمان طولانی ممکن است طول بکشد باید تمامی کاربرها از بانک خارج شوند بعد data base خاموش می شود. (یعنی خودش کاربرها را

خارج نمی کند، صبر می کند همه خارج شوند بعد data base را shut down می کند)

نکته: در تمامی روش ها از زمانی که دستور shut down را وارد می کنیم از ورود کاربر جدید ممانعت به عمل می آورد.

Shut down transactional :

صبر می کند transaction ها commit شوند یا خودشان را toll back کنند سپس کاربرها را خارج می کند data base را shut down می کند

RMan3

Shut down immediate; ← صبر می کند transaction هایی که نزدیک به commit شدن هستند commit شوند، transaction های بزرگ را roll

back می کند و کاربرها را خارج می کند.

اون هایی که در pending queue transaction هستند commit می شوند اونایی که در pending queue transaction هنوز نیافتادند اونها را roll back می

کند. چون در سطح transaction ما چند نوع صف داریم و آخرین صف (نهایی) pending است.

Shut down aboard هیچ فرقی با کشیدن برق کامپیوتر یا stop کردن سرویس oracle در ویندوز ندارد.

Oracle RDBMS\ product\ 11.2.0\ db home-1\ dbs\ init.ora

Init. ora فایل parameter است.

پس:

Shut down immediate;

Start up mount;

ابتدا data base می رود تو وضعیت Start up no mount بعد می ره به وضعیت Start up mount. بعد archive log وقتی می خواهیم back up بگیریم حتماً باید از Redo file های من oracle کپی داشته باشد. Redo file ها به صورت sequence کار می کنند و دوباره برمی گردد Redo اول را از بین می برد. برای back up گیری باید oracle را موظف کنیم این Redo اول را از بین نبرد. یواشکی جایی کپی کند آن را.

Alter data base archive log;

اگر data base open باشد نمی توانیم این دستور را اجرا کنیم.

دلیل این امر: چون ممکن است transaction انجام شود Redo تحت تأثیر قرار بگیرد و موقعی که Redo دارد use می شه نمی توانیم استفاده کنیم. مثلاً الان می خواهیم بدانیم data base در چه mode ی است؟ از static data dictionary استفاده می کنیم.

Describe v\$ data base;

بسیار مهم است و مقادیر زیادی هم در داخل دارد.

فیلد name و log mode را بگیریم.

Select loge-mode from v\$ data base;

	Log-mode
	ARCHIVE Elog
خروجی	مود log را به ما برمی گرداند

Alter data base no archive log;

Select log- mode from v\$ data base;

Log-mode
NO ARCHIVE log

از این دیتا دیکشنری استفاده های دیگری هم می شود، flash back mode و زمان ساخت دیتابیس، زمان ساخت مخزن و

RMan4

حالا به همان حالت قبلی بازمی گردیم یعنی

alter data base archive log;

exit;

تا حالا یک table t1 در داخل sql ایجاد کردیم. به package ی که به صورت free در داخل data base اوراکل قرار داده شده به صورت فایل mdd. به

صورت rman (recovery manager) :

Rman target /

دستور rman را صادر می کنیم.

(برای رفتن به محیط RMAN)

RMAN/ back up data base plus archive log;

این محیط این قابلیت را دارد که از خارج از سیستم عامل راه اندازی شود.

data base باید در وضعیت open باشد.

RMAN/ Start up;

RMAN/ back up data base plus archive log;

برای back up گیری و اینکه در بک آپ archive log هم بیاید از این دستور استفاده می شود.

این back up گیری online است. RMAN یک سری روش های دیگر دارد که recovery catalog (rc) گویند.

حال می بینیم فایل هایی که ذخیره کردیم در مسیر set back up\ oracle\ back up\ ORDBMS\ ذخیره شد.

Shut کنیم. Shut down را data base می کنیم این است که sql plus را حذف می کنیم. اولین کاری که در محیط T1 جدول sql pl حال در محیط
down immediate;

Start up;

Select * from t1;

Drop table t1;

Shut down immediate;

حالا data file ها را هم حذف می کنیم.

D\ Root\ oracle RDBMS\ oracle data\ orc\

(در واقع داریم بدترین اوضاع را شبیه سازی می کنیم)

در اینجا هم data file ها را پاک کنیم توجه داشته باشید به هیچ عنوان فایل CTL فایل را نباید پاک کرد و بهتر است در یک فایل دیگر یک کپی از آن
بگیریم.

حالا از محیط sql plus خارج و وارد محیط RMAN می شویم.

RMAN> Start up mount;

Start up mount کنترل فایل ها را Read و Open می کند و data file ها را open می کند.

اما الان data file ی وجود ندارد تا open کند.

Restore data base until time “to date (‘20130124115400’ , ‘yyymmddhh24miss’)”

به تاریخ قبل از delete شدن جدول برمی گردیم که مصادف است با تاریخی که back up را گرفتیم.

RMAN 5

حال می رویم که start ← D ← root ← ... ← oracle ← data file ها همه restore شدند (هنوز recover نشده اند) restore : فایل ها برگردانده شده
ولی instance هنوز از آن ساخته نشده است.

Recover data base until time ‘to-date (‘20130124235600’ , ‘yyymmddhh24 miss’)”

Exit;

Sql plus/ assysdba

Shut down immediate;

Start up mount;

Alter data base open reset logs;

Select * from t1;

هر back up دو تا point دارد point آخرین لحظه ای که data base بوده است و point (before point) در زمان قبل تر است (data file) ها بودند).

Redo : کل وضعیت های data base را دارد.

پس یک در تاریخ ۲۵ سیستمی back up گرفتیم. ❶

دوباره تاریخ را به ۲۴ تبدیل کردیم و دوباره back up گرفتیم. ❷

زمانی که با دستور restore خالی (بدون تاریخ) را رد می کنیم می رود آخرین back up را برمی گرداند (۲۵ رو برمی گرداند).

بعد که می خواهیم بگوئیم recover کن روی تاریخی قبل از تاریخ ۲۵ ام پاسخ می دهد که نمی شود.

RESET LONGS option only valid after an incomplete data base recovery

مبحث *import, export* و *back up* گیری

Oracle برای back up گیری و export گیری چندین روش را ارائه می کند.

پشتیبانی اطلاعات بر مبنای دو حالت است physical و logical (فیزیکی و منطقی)

یا اصطلاحاً:

- پشتیبانی با ساختار (فیزیکال): دو روش وجود دارد classic and data pump
- پشتیبانی بی ساختار (logical): روش های متعددی وجود دارد مهم ترین روش استفاده از یک روش به نام recovery manager است یا

محیط Rman

پشتیبانی با ساختار هم برای back up گیری هم برای انتقال مورد استفاده قرار می گیرد با همان دو روش classic و data pump در back up گیری بی ساختار توجه ما در انتقال داده نیست و لزوماً بحث back up مطرح است. و نمی توانیم این back up را روی یک سیستم دیگر restore کنیم و این back up فقط مربوط به همان محیط است.

یک آپ گیری با ساختار در جلسه آینده مورد بررسی قرار خواهد گرفت.

Rman ذاتاً back up بی ساختار است. وقتی back up می گیریم حتماً باید در همان کامپیوتری باشد که control file ها در آن موجودند. در چه شرایطی back up می گیریم. back up می گیریم که اگر DF فایل هامون خراب شد برگردد، اگر رکوردهامون اشتباهاً insert و delete شد برگردد.

ساختار data file ها در داخل control file ها است از control file، Rman، back up نمی گیرد.

البته می توانیم آن را مجبور کنیم که back up بگیرد اما بهتر است زمانی که می خواهد back up بگیرد برویم روی روش باساختار.

اگر جدول t1 را حذف کنیم در زمان restore کردن این table برخواهد گشت. این روش برای صدمه دیدن data خوب است البته راه برگشت table هست ولی ما از RMan معمولاً به این روش استفاده نمی کنیم و برای بازیابی اطلاعات از Rman استفاده می شود.

برای انتقال داده ها از export استفاده می کنیم که دو روش دارد:

۱. classic export

۲. data file export

فرق پارامتر فایل و کنترل فایل: پارامتر فایل، پارامترهای اولیه را set می کند ولی control فایل مسیر اولیه را نگه داری می کند.

روش exp

طریقه انجام export گیری: وارد محیط cmd می شویم

تذکر- وارد محیط sql plus نمی شویم (cmd)

دستور exp را تایپ می کنیم

که یک dop file به ما می دهد و این dop را می توان به عنوان back up کامل data base از آن schema ی مد نظر ماست. یعنی از view و table و sequence و back up می گیرد.

هنگام تایپ exp از ما user و pass می پرسد.

User name: system wxe

Password: myjava123

عددی که می بینیم (۴۰۹۴) مقدار buffer size است که به توصیه oracle نباید تغییر کند.

Buffer size زمان export گیری را مشخص می کند و RAM بیشتری از ما می گیرد و اگر مقدار آن را کم کنیم زمان بسیار بیشتری از ما می گیرد شاید هم on time out شویم.

D:\sami.DMP

را تایپ می کنیم تا یک فایل به نام Sami.DMP را برآیمان ایجاد کند. از ما می پرسد از چه چیزی می خواهی back up بگیری؟

(1) E(nder data base), (2) U(sers), or (3) T(ables):

از table یا user یا data base می خواهی back up بگیری؟

User ی با نام amirsam با یک جدول به نام t1 مفروض است.

حال می خواهیم از user، back up بگیریم. * همچنین بهتر است همواره از user، back up بگیریم.

سوالاتی که از ما می پرسد:

۱. Export کند data table های آن را؟

۲. Export کند grant های آن را؟

۳. Compress کند خروجی آن را؟

۴. User ی که می خواهی export بگیری ا اعلام کن؟ amirsam

۵. User بعدی که می خواهی export بگیری را اعلام کن؟ مثلاً sys

تا مادامی که پاسخ این سؤال را وارد کنیم مجدداً آن را از ما می پرسد. هنگامی که جواب آن را خالی رد کنیم عمل export گیری انجام می شود از user های که لیست آنها را وارد کرده ایم.

Export terminated successfully without warning.

بدون هیچ error ی انجام شد.

به مسیر D:\ می رویم. این فایل back up را باید حفظش کنیم.

به محیط sql plus برمی گردیم و user را Drop می کنیم.

Drop user amirsam cascade;

و دوباره user را ایجاد می کنیم

Create user amirsam identified by my java123;

- حتماً user باید ایجاد شود (هنگام
- اگر از data base، بک آپ گرفته بودیم باید فقط data base را ایجاد کنیم.
- در table باید table ایجاد شود.

حال در همان محیط imp را تایپ می کنیم. Username و password را وارد می کنیم آدرس DMP فایل را می خواهد که وارد می کنیم
D:\sami.DMP سپس buffer size را می پرسد که آن را رها می کنیم/

سؤال اول

Ignore create error ? No

Import grants? Yes

Import data table? Yes

Import user نام amirsam ?

حال اگر وارد کنیم

Select * from amirsam.t1

مقادیر آن را می بینیم

روشی که تا کنون مطرح بود کلاسیک exp است و در حال حاضر استفاده نمی شود. این روش، روش مساعدی نیست زیرا که buffer size ی که از ما می خواهد به صورت static است و در هر شرطی این buffer size باید پایدار بماند.

اگر buffer size را بسیار کوچک کنیم که به کارهای دیگر لطمه وارد نکند.

ممکن است back up گیری زمان طولانی طول بکشد.

یا اگر آن را بزرگ در نظر بگیریم ممکن است مشکلی پیش بیاید. در زمانی که back up می گیرد تراکنش روی سرور زیاد باشد out of memory می کند و data base پائین می آید. وقتی data base out of mem می شود اولین اتفاقی که می افتد این است که oracle ← tms listiner را stop می کند که البته در ورژن IIgr2، tms lisiner ، stop نمی شود.

روش دوم: data pomp

یک. سرعت exp را به سرعت بالا برده است زیرا buffer size را در این حالت dynamic می کند.

به این معنا که ماکزیمم بافر سایز را می گیرد وقتی فشار روی بافر سایز می یاد آن را به صورت اتوماتیک هنگام export گیری پائین می آورد و فشار که از بین رفت دوباره آن را بالا می برد.

که این روش از log به بعد وجود دارد.

در EXPDP، cmd را تایپ می کنیم user را systemwx و pass را my java 123 را وارد می کنیم هیچ سؤالی از ما نمی پرسد و شروع به back up گیری می کند. از current user ی که login کرده است back up می گیرد. و اتوماتیک مسیر را خودش تعیین می کند.

این روش دو مشکل اساسی دارد

۱. وقتی base با اون کار می کنیم از ما هیچ سؤالی نمی پرسد و باید به صورت پارامتریک به آن پاس بدهیم. وقتی این کار را مجدداً انجام دهیم

error می گیریم چون اون Dmp فایل در همان مسیر که خودش تعیین کرده وجود دارد. و مسیر را خود oracle را مشخص می کند.

روش معکوس آن IMPDP :

IMPDP را در cmd تایپ می کنیم username را وارد می کنیم که البته به علت وجود داشتن table1 و user amirsam به ما error می دهد (یا از کنار دستورمان می گذرد و اجراش نمی کند).

پس اول آن user را drop می کنیم و مجدداً آن را ایجاد می کنیم (بدون تعریف جدول t1)

Create user amirsam identified by my java 123;

Create DBA, connect to Amirsam;

سپس تایپ دستور IMPDP را تایپ می کنیم و تست می کنیم.

Select * from amirsam.t1;

که البته در این حالت به user هم نیاز نداریم و می توانیم آن را هم پاک کنیم.

Batch file چیست؟

Batch file فایلی با پسوند bat است در سیستم عامل های مختلف با پسوندهای مختلف است مثلاً sh (از shell)

در Batch file می توانیم command هایی بگذاریم.

به درایو C برو صفحه را تمیز کن DIR را به من بده.

C:

CLS

DIR

PUASE

این command ها فرامین سیستم عامل windows است در این سیستم عامل ما فرامین را با نام batch می شناسیم. در linux با نام shell

command می شناسیم و (DIR در linux لیست می شود یعنی LS). و طبعاً مسیره های ها در linux متفاوت است.

حالا ما با همین Batch file یک حرکت انجام می دهیم.

دستور expdp یک سری پارامتر دریافت می کند.

Expdp %AdminName% / %ADminpass% w % REPOSITORY %

Amirsam : Admin Name

my java 123 : Admin pass

XE : Repository

Schemas= % schema% dump file-\$ % schema% \$ % date : ~ 4, 2%- %date :~ 7, 2%- %date: ~ 10, 4%\$.dmp

این را با تاریخ امروز سیستم جایگزین می کنیم.

Amirsam : Schema

نکته: schemas یکی از دستورات ورود به محیط expdp است که می توانیم در جلوی آن نام چندین user که قرار است از آنها back up گرفته شود

(مثل Akbar ، Asghar ، Amirsam و ...) را قرار دهیم.

نکته: dump file آدرس dmp file ماست.

که ما در اینجا اسم dump file را تاریخ امروز سیستم گذاشتیم که با قرار دادن آن بین \$ ---\$ به وجود می یاد.

Explorer % ORACLE-HOME-DIRECTORY % \app\oracle\ admin\ % REPOSITORY % \dpdump

یک دستور ویندوزی است که پنجره ای را که مسیر ما در آن است را با فایل با پسوند dpdump را باز می کند.

و نهایتاً pause

معمولاً الان back up را ساعتی نمی گیرند چون با flash back قابل انجام است. و به صورت روزانه به این روش انجام می شود.

oracle خودش دارای یک scheduler (زمانبند) است که یک ایراد دارد که اگر oracle sleep شود آن نیز sleep می شود پس بهتر است در سرورهای عظیم هم از زمانبند سیستم عاملی استفاده کنیم.

ویندوز هم خودش دارای یک scheduler است.

Accessories → Task scheduler → Create Basic task →

: Recycle Bin

پاک کردن محفظه سطل آشغال.

اشیا بعد از حذف داخل سطل آشغال می روند.

از اینجا به بعد در فیلم فقط voice واضح بود و تصویر نداشت.

“Form Developer & Report Builder”

نصب Developer Suit (مطرح کردن نکات)

دیسک ۱ ← set up ← Next ← مسیر : path نام و

در سربرگ select installation type گزینه complete را انتخاب می کنیم. ← Next

اگر روی سیستم iis نصب کرده باشیم در سربرگ provide outgoing mail server آن را وارد می کنیم. هنگام نصب حتماً باید دو مسئله را رعایت کنیم.

در run تایپ کنیم regedit , msn language را حتماً set می کنیم که مربوط به multi language support می شود که اگر دیتابیس را نصب می کنیم حتماً باید فارسی نصب نشود باید آن را روی American-America AR8MSWIN1256 تنظیم می کنیم. که NLS عربی در آن است و فارسی را هم support می کند.

Developer suit بر اساس Default language. Code page را set می کند.

Local machine → software → oracle → Developer suite Home 1 →

Nls را روی nls فارسی set می کنیم.

همچنین پس از نصب Developer suite باید connection آن را به data base برقرار کنیم. به این صورت که پس از نصب start ← Dev-suit Home

oracle-1 ← Next Configuration Assistant ←

در اینجا می خواهیم به data base معرفی کنیم از کدام data base استفاده کنیم. (connection روی listener روی پورت ۱۵۲۱) و انتخاب گزینه

Local Net Service Name Configuration ← Next ← Reconfigure یا گزینه Add (اگر قبلاً آن را نصب نکرده باشیم ADD را انتخاب می

کنیم) ← نام سرویس (اسم بانک اطلاعات) ← پورتکل ← local host ← ۱۵۲۱ ← Test ← ← finish

PI/SQL Developer : نرم افزاری ویژوال است به جای استفاده از sql plus

در قسمت All object, function ها و table ها و view ها را برآیمان می آورد.

در ان صورت تمام Object های سیستمی و غیر سیستمی را نشان می دهد ولی اگر بخواهیم تنها object های خودمان را ببینیم گزینه my object را

انتخاب می کنیم.

اگر روی هر table ی کلیک راست کنیم به یک سری کارها دسترسی پیدا می کنیم.

مثلاً Query Data

- می توان به اطلاعات جدول دسترسی پیدا کرد.
- Edit Data قابلیت ویرایش اطلاعات را هم همین جا به ما می دهد.
- می توانیم در این محیط از object هایمان Export بگیریم.
- می توانیم ارتباطات بین table ها را با استفاده از Diagram Window مشاهده کنیم.
- در کنار پنجره دکمه ای به نام Execute Query داریم که می توانیم با زدن آن Query نوشته شده توسط خودمان را اجرا کنیم.

در محیط Report Builder به دو صورت می توانیم طراحی انجام دهیم یا به کمک wizard خودش یا به صورت manual .

حالت Report wizard :

انواع Style به صورت: ماتریسی، گروهی و انواع wizard ها

→ Data source type → tools انتخاب → Report wizard → style , title

نوشتن Query برای گرفتن اطلاعات از دیتابیس و دادن user و pass با استفاده از دکمه connected

→ table → فیلدهای کردن → filter انتخاب اسم فیلدها برای هر ستون → finish → template

حال در محیط خود report builder اطلاعات را به ما نشان می دهد.

در این محیط آیکن به اسم data module داریم که در آن Query هامون را به آن می دهیم.

در این محیط آیکن به اسم web source داریم که در آن بخش jsp آن است که آن را generate می کند.

در قسمت layout چند گزینه داریم config off (همبستگی فیلدها از بین می رود و از فیلدها را جدا جدا باهاشون کار می کنیم. از یک frame و یک

repetition frame تشکیل شده است فیلدهای تکرار شده در داخل یک

در گزارشی که در حال حاضر می بینیم ما یک سری از فیلدها را داریم (به طور مثال Emp-ID و Emp Name) و می خواهیم یک سری دیگر از فیلدها

را نیز برای نشان دادن انتخاب کنیم از گوشه پنجره روی گزینه sql Query کلیک می کنیم و یک Query جدید در اینجا اضافه می کنیم حال می

توانیم به کمک یک link (که آیکن آن در نوار ابزار است) job-ID را به job-ID جدول جدید متصل کنیم و گزارش ها را به هم وصل می کنیم.

برای استفاده از title می توانیم از آیکن های frame و repetition frame استفاده می کنیم.

و با کلیک راست روی آن repetition frame در قسمت source گروه را انتخاب می کنیم.

حال اگر گزارش را run کنیم (دو حالت داریم run paper layout و run web layout یعنی خروجی را به صورت paper یا web می خواهید).

یکی دیگر از خاصیت های data module بر اینکه می توانیم در select آن Group By داشته باشیم در سطح data module می توانیم یک سری

کارهای دیگر هم بگذاریم مثلاً بر اساس job ID گزارش را گروه بندی کنیم.