



ORACLE OBJECT MANAGEMENT



اسکیما :

هر وقت هر کاربری مالک شی در بانک اطلاعاتی باشد ماهیت کاربر به اسکیما تغییر داده می‌شود، هر اسکیما از یک یا چندین شی درست شده است هر اسکیما می‌تواند شامل اشیاء ذیل باشد :

- Procedure
- Function
- Package
- Trigger
- Table
- Cluster
- Index
- View
- Sequence
- Synonym

کاربری که مالک یک شی است می‌تواند دسترسی شی مورد نظر را به کاربران دیگر اعطا کند.

قوانین نام گذاری اشیاء:

- ✓ حداکثر ۳۰ کاراکتر
- ✓ کلمات کلیدی و رزرو شده نباشد
- ✓ باید با A تا Z یا a تا z شروع شود
- ✓ می‌توان مجموعه ای از حروف، اعداد، #، _ و \$ باشد

نکته: اگر از حروف کوچک در نام گذاری اشیاء استفاده شود بصورت اتوماتیک بزرگ خواهد شد، اگر نام شی در " قرار گیرد نام آن بصورت اتوماتیک بزرگ نخواهد شد

فضای نام :

نام اشیاء موجود در هر اسکیما باید یکتا باشد ولی اشیاء می‌توانند در اسکیماهای مختلف دارای نام یکسانی باشند، عنوان مثال جدول EMPLOYEES می‌تواند در هر دو اسکیمای HR و HS وجود داشته باشد

انواع داده‌ای (DataTypes):

۱- کاراکتری

مشخص کننده Character Set: زبانی است که DB (پایگاه داده) به هنگام ذخیره و بازیابی از آن استفاده می‌کند و هنگام نصب مشخص می‌شود

- Char(size) ✓
- nchar(Size) ✓
- varchar2(Size) ✓
- nvarchar2(Size) ✓
- raw ✓ ← برای ذخیره مقادیر کاراکتری به صورت باینری استفاده می شود

۲- عددی

- Number(precision , scale) ✓
- precision ○ ← بیشترین عدد صحیح که می تواند ثبت شود
- scale ○ ← بیشترین عدد اعشار که می تواند ثبت شود
- integer ✓ ← برای نگهداری مقادیر صحیح استفاده می شود.

۳- زمان و تاریخ

- date ✓ ← امکان تغییر شکل مقادیر تاریخ با استفاده از nls_date_format وجود دارد و برای اینکار می توان از to_char و to_date استفاده کرد.
- timestamp(n) ✓ ← مانند date است با این تفاوت که تا کسری از ثانیه را مشخص می کند که عددی بین ۰ تا ۹ است و پیش فرض آن عدد ۶ می باشد.
- timestamp with timezone ✓ ← به غیر از timestamp اختلاف زمان بین UTC و زمان محلی را نیز بر حسب ساعت و دقیقه نگهداری می کند.
- timestamp with local timezone ✓ ← مانند نوع داده ای قبل است با این تفاوت که اختلاف زمان بین UTC و زمان محلی در پایگاه داده ذخیره نمی شود بلکه هنگام خوانده شدن او را اختلاف زمان را بر حسب timezone کاربر باز می گرداند.
- interval year(y_precision) to month ✓ ← بازه زمانی از سال تا ماه را نگهداری می کند
- y_precision تعداد ارقام سال را مشخص می کند که پیش فرض ۲ است.
- interval day(d_precision) to second(s_precision) ✓ ← بازه زمانی روز تا ثانیه را نگهداری می کند
- d_precision تعداد ارقام روز را مشخص می کند

○ s_precision تعداد ارقام ثابته را مشخص می کند که از ۰ تا ۹ مقدار دهی می شود

انواع داده های بزرگ :

- 4 GB ← Clob ✓
- 4 GB ← NClob ✓
- 4 GB ← BLOB ✓
- 4 GB ← BFile ✓
- 2 GB ← Long مقادیر کاراکتری تا حداکثر 2 GB ✓
- 2 GB ← Long raw مقادیر کاراکتری به صورت باینری حداکثر 2 GB ✓

تخمین اندازه یک جدول

۱- روش OEM

برای استفاده از این روش باید وارد مسیر ذیل در محیط EM (Enterprise Management) شده و در صفحه باز شده تعداد رکوردهای احتمالی را وارد کرده و دکمه estimate table size را انتخاب کنید.

schema\database objects\tables\create\continue\estimate table size

۲- با استفاده از پکیج dbms_space و با استفاده از رویه create_table_cost

محدودیتها (constraint)

داده های ثبت شده را در سطح پایگاه داده صحت سنجی کرده و از ذخیره داده های نامعتبر جلوگیری می کند ، اطلاعات constraint ها در user_constraint ذخیره می شود و به دو روش تعریف می شود

۱- در سطح فیلد

۲- در سطح جدول ← اگر بخواهیم بر روی بیش از یک فیلد تعریف شود از این روش استفاده می کنیم

انواع محدودیتها (Constraints)

- ✓ Not null
- ✓ Unique
- ✓ Check
- ✓ Foreign key

✓ Primary key

ایجاد محدودیت (constraint) پس از ایجاد جدول

در صورتی که PK و UK هنگام ایجاد جدول تعریف شود یک index متناظر برای آن ایجاد می گردد

حذف محدودیت :

با استفاده از دستور ذیل می توان محدودیتی را که بر روی جدول وجود دارد را حذف کرد

```
alter table <table_name> drop constraint <constraint_name>;
```

به دلیل اینکه هر جدول فقط یک PK دارد می توان از دستور ذیل برای حذف PK (کلید اصلی) استفاده کرد

```
alter table <table_name> drop primary key;
```

میتوان با استفاده از دستور ذیل PK یک جدول را به همراه تمامی Fk هایی که به آن اشاره می کند را حذف کرد

```
alter table <table_name> drop primary key cascade;
```

به صورت اتوماتیک با حذف PK و UK تمامی ایندکس هایی که بر روی فیلدهای مذکور وجود دارند حذف خواهد شد مگر اینکه از دستور Keep Index استفاده کنید.

```
alter table <table_name> drop primary key keep Index;
```

تغییر نام محدودیت :

```
alter table <table_name> rename <old_constraint_name> to <new_constraint_name>;
```

فعال یا غیر فعال کردن محدودیت ها :

1- alter table <table_name> add constraint <constraint_name>

enable | disable

validate | novalidate ;

2- Alter table <table_name> Add constraint <constraint_name>

enable | disable

validate | Novalidate

constraint <constraint_name> ;

Enable Validate ✓

محدودیت برای تمام داده‌های موجود و داده‌های قبلی اعمال می‌شود

Enable Novalidate ✓

محدودیت برای داده‌های جدید اعمال شده ولی برای داده‌های قدیمی اعمال نمی‌شود

Disable Validate ✓

محدودیت برای داده‌های قدیمی اعمال شده و برای داده‌های اعمال نمی‌شود

Disable Novalidate ✓

محدودیت برای داده‌های قدیمی و جدید اعمال نمی‌شود

کنترل محدودیت با تعویق

می‌توان بررسی شدن یه محدودیت (constraint) را تا زمان commit به تعویق انداخت برای این کار می‌توان گزینه‌های deferrable و deferred را انتخاب کرد.

ایجاد جدول با استفاده از (CTAS) Query

یک جدول می‌تواند بر اساس خروجی یک پرس و جو ایجاد شود

```
create table <table_name> as
select * from employees;
```

اصلاح جدول

پس از ایجاد جدول می‌توان با استفاده از عملیات ذیل آن جدول را تغییر داد

✓ اضافه کردن یک یا چند ستون

✓ تغییر ویژگی ستون‌های جدول

✓ اضافه کردن توضیحات برای جدول یا ستون

✓ حذف داده‌های جدول

✓ تغییر وضعیت جدول به حالت فقط خواندنی (read only)

```
alter table <table_name> read only;
```

✓ تغییر وضعیت جدول به حالت خواندنی و نوشتنی (read write)

```
alter table <table_name> read write;
```

- ✓ تغییر نام جدول
- ✓ حذف جدول

جداول موقت

جداول موقت به دو نوع تعریف می‌شوند

- 1- per transaction
- 2- per session

create global temporary table <temp_table_name>
(column definition) On commit delete | preserve rows;

محدودیت‌های جداول موقت شامل موارد ذیل می‌باشد

- ✓ نمی‌توانند مبتنی بر **index** باشد
- ✓ نمی‌توانند پارتیشن بندی یا **cluster** بندی شوند
- ✓ کلید خارجی (**foreign key**) ندارد
- ✓ نمی‌توان ستونی از نوع **varray** یا **nested table** در آنها تعریف کرد.
- ✓ همواره در **tablespace** موقت (**temp tablespace**) ایجاد شده و امکان تغییر **TS** آن وجود ندارد.
- ✓ نمی‌توان بر روی آنها **DDL** اجرا کرد
- ✓ نمی‌توان داده‌های آن را **export** کرد

ویژگی‌های جداول موقت

- ✓ هر **session** داده‌های مربوط به خود را می‌بیند
- ✓ با از بین رفتن **session** داده‌های مربوطه حذف می‌شود حتی اگر **session** بصورت غیر عادی خاتمه یافته باشد.
- ✓ با حذف داده‌های جدول موقت فقط داده‌های **session** جاری حذف می‌شود.
- ✓ می‌توان برای جداول موقت **index** تعریف کرد
- ✓ می‌توان از پرس و جوهایی که ترکیب جداول موقت و دائم است استفاده کرد.
- ✓ جداول موقت می‌توانند **trigger** داشته باشند

DD (Data Dictionary) های مربوط به جداول

DBA_tables
DBA_table_columns
Describe <table_name>;

DD (Data Dictionary) های مربوط به محدودیت‌ها (Constraints)

DBA_constraints

DBA_Cons_Columns

پارتیشن بندی جداول

با استفاده از این امکان که از اوراکل 8i به بعد اضافه شده است می توان جداول، ایندکس ها و جداول مبتنی بر ایندکس را پارتیشن بندی کرد.

مزیت پارتیشن بندی جداول :

- ✓ افزایش سرعت اجرای پرس و جوها
- ✓ مدیریت آسان جداول بزرگ با تقسیم آنها به جداول کوچکتر
- ✓ انجام سریع عملیات Export و Import و انتقال داده ها مابین بانک های اطلاعاتی
- ✓ کاهش زمان Recovery

: Partition Pruning

هنگام اجرای پرس و جو در جدولی که پارتیشن بندی شده است فقط پارتیشن هایی مرتبط جستجو انجام می شوند که به اینکار Partition Pruning می گویند.

Partiton key

فیلد یا فیلدهایی که براساس آن جدول پارتیشن بندی می شود.

چه زمانی باید جدول را پارتیشن کرد

- ✓ حجم جدول از 2 GB بیشتر باشد
- ✓ جدولی که داده های آن باید آرشیو شود
- ✓ وقتی نیاز است تا اطلاعات جدول بر روی چندین دیسک توزیع شود

نکته: می توان جدول یا پارتیشن های جدول را فشرده سازی کرده و سپس در دیسک ذخیره کرد.

روش های پارتیشن کردن جداول

نکته: نمی توان جداولی که دارای ستون های Long یا Long Raw هستند را پارتیشن بندی کرد.

روش های پارتیشن کردن جدول

۱- ساده

- List ✓
- Range ✓
- Hash ✓
- Interval ✓
- System ✓
- Refrence ✓
- Virtual ✓

۲- ترکیبی

- List-Range ✓
- Hash-Interval ✓
- ... ✓

Interval

- ✓ فقط براساس ستونی پارتیشن بندی انجام می شود که نوع ستون **Date** یا **Number** باشد .
- ✓ برای ایجاد پارتیشن **Interval** باید یک پارتیشن ابتدایی برای آن مشخص شود.

: System

انجام عملیات DML بدون مشخص کردن پارتیشن هدف امکان پذیر نخواهد بود.

: Virtual

پارنیشن بندی می تواند براساس ستونی که از ترکیب ستون های دیگر بدست می آید انجام شود.

DD (Data Dictionary) های مربوط به پارتیشن

DBA_tab_partitions
 DBA_part_tables
 DBA_ind_partitions