

ওরাকল ডাটাবেজ অ্যাডমিনিস্ট্রেশন

মো: ইফতেখারুল আলম

(পূর্ব বক্রাণিতের পর)

গত পর্বে আমরা ওরাকল ডাটাবেজ ব্যবস্থাপনার সবচেয়ে গুরুত্বপূর্ণ অংশ ডাটাবেজের অন্যতম প্রধান দুইটি উপাদান কন্ট্রোল ফাইল এবং রিডু লগ ফাইল নিয়ে আলোচনা করেছি। এই পর্বে আলোচনা করা হয়েছে ডাটা ফাইল নিয়ে।

ডাটা ফাইল এবং টেবিলস্পেস

ওরাকল লজিক্যাল টেবিলস্পেসে এক ফিজিক্যাল ডাটা ফাইলে ডাটা স্টোর করে। স্বল্পত ডাটা ফাইল, টেবিলস্পেস এবং ডাটাবেজ প্লানার একই কন্ট্রোল মনে হলেও তাদের মধ্যে কিছু গুরুত্বপূর্ণ পার্থক্য রয়েছে। ওরাকল ডাটাবেজ এক বা একাধিক লজিক্যাল স্টোরেজ ইউনিট বা টেবিলস্পেসের সমন্বয়ে গঠিত। এই টেবিলস্পেস আবার এক বা একাধিক ফাইলের সমন্বয়ে গঠিত। এই ফাইলগুলো হচ্ছে ফিজিক্যাল ক্যাটামে, যা অপারেটিং সিস্টেমের মাধ্যমে নিশ্চিত করা হয়। এই ফাইলকে বলা হয় ডাটা ফাইল। মূলত ডাটাবেজের ডাটা সাংগঠনিক প্রক্রিয়ায় ডাটা ফাইলে সংরক্ষিত হয় এবং এই ডাটা ফাইলের সমন্বয়ে তৈরি হয় টেবিলস্পেস। ওরাকল ডিবিএ এই টেবিলস্পেসের মাধ্যমেই ডাটা ফাইলকে ম্যানেজ করে থাকে।

টেবিলস্পেসের প্রকারভেদ

ডিবিএ টেবিলস্পেস তৈরি করে যাকে ডাটাবেজকে সহজে নিয়ন্ত্রণ এবং মেনেটাইন করতে পারে। ওরাকল সার্ভার দুই ধরনের টেবিলস্পেস ধারণ করে- ০১, সিস্টেম ০২, ননসিস্টেম।

সিস্টেম : সিস্টেম টেবিলস্পেস ডিবিএকে আলোচনা করে তৈরি করতে হয় না, ডাটাবেজ তৈরির সময় এটা তৈরি হয়ে যায়। ডাটা ডিকশনারি ও সিস্টেম আয়ু সেগমেন্ট এই টেবিলস্পেসের অধীন থাকে। অবশ্যই একটা ডাটাবেজের সিস্টেম টেবিলস্পেস পাকা বাধ্যতামূলক। এতে ইউজারের ডাটা কোন কখন স্টোর হয়ে থাকে, তবে না করাটাই ভালো।

ননসিস্টেম : এখানেই মূলত থাকে ইউজারের সংরক্ষিত ডাটা। ননসিস্টেম টেবিলস্পেসের কারণে ডিবিএ ডাটা ফাইলকে ডাটাবেজের ম্যানেজ করতে পারে।

টেবিলস্পেস তৈরি করার কমান্ড :

```
CREATE TABLESPACE userdata
DATAFILE 'u01/oradata/userdata01.dbf'
SIZE 5M;
```

স্পেস ব্যবস্থাপনা : টেবিলস্পেসে এক্সটেন্ড হিসেবে স্থান বরাদ্দ করে। দুইটি পদ্ধতিতে টেবিলস্পেস বর্ধিত অথবা ব্যবহার হওয়া স্থানকে ম্যানেজ করে থাকে। ০১, লোকালি ম্যানেজড ০২, ডিকশনারি ম্যানেজড।

লোকালি ম্যানেজড : এক্সটেন্ড এই পদ্ধতিতে বিটম্যাপের মাধ্যমে ম্যানেজ হয়ে থাকে। যখন কোনো এক্সটেন্ড বরাদ্দ অথবা স্থান হ্রাস তখন ওরাকল সার্ভার বিটম্যাপের মন পরিবর্তন করে দেয়, যার ফলে ডাটা ব-কোর বর্তমান অবস্থা

```
CREATE TABLESPACE userdata
DATAFILE 'u01/oradata/userdata01.dbf'
SIZE 500M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 256K;
ডিকশনারি ম্যানেজড টেবিলস্পেস তৈরির কমান্ড :
CREATE TABLESPACE userdata
DATAFILE 'u01/oradata/userdata01.dbf'
SIZE 500M
EXTENT MANAGEMENT DICTONARY
DEFAULT STORAGE ( initial 1M NEXT 1M );
```

লোকালি ম্যানেজড টেবিলস্পেস যে কারণে ডিকশনারি ম্যানেজড টেবিলস্পেস অপেক্ষা ভালো

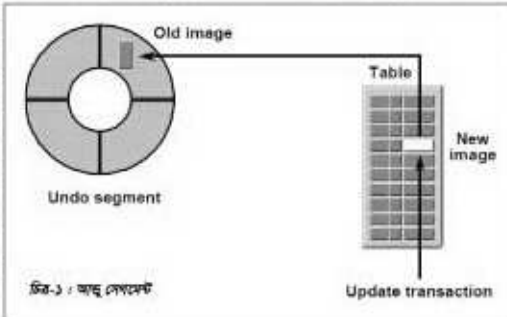
লোকালি ম্যানেজড টেবিলস্পেস ডিকশনারি ম্যানেজড অপেক্ষা বেশ কার্যকর। কারণ-
০১, লোকালি ম্যানেজড টেবিলস্পেস এক্সটেন্ড স্বয়ংক্রিয়ভাবে শূন্যস্থান পূর্ণা করে ফেলে, যা ডিকশনারি ম্যানেজড টেবিলস্পেসে সম্ভব নয়।
০২, ডাটা ডিকশনারিতে কোনো ট্রিপ্পলের তথ্য সংরক্ষণ করতে হয় না।

০৩, লোকালি ম্যানেজড টেবিলস্পেস রিকারসিভ স্পেস ব্যবস্থাপনার মাধ্যমে কাজ করে না, যার ফলে এক্সটেন্ডে স্থান বরাদ্দ অথবা ডাটা ডিকশনারি টেবিল পূর্ণ করার জন্য নতুন কোনো অপারেশন করতে হয় না।
০৪, এক্সটেন্ডের আকার স্বয়ংক্রিয়ভাবে নিয়ন্ত্রণ হয়ে থাকে।

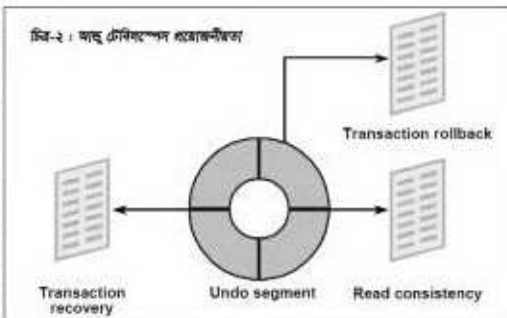
ডিকশনারি ম্যানেজড টেবিলস্পেস লোকালি ম্যানেজড-এ পরিবর্তনের কমান্ড :
DBMS_SPACE_ADMIN.
TABLESPACE MIGRATE
TO LOCAL('SYSTEM');

আয়ু টেবিলস্পেস : যখন ডাটাবেজ ডাটার কোনো পরিবর্তন হয়, তখন পরিবর্তন পূর্ববর্তী ডাটা ওরাকল যে সেগমেন্টে সংরক্ষণ করে থাকে বলা হয় আয়ু সেগমেন্ট। (চিত্র-১)

এই আয়ু যে টেবিলস্পেসের আওতাধীন থাকে তাকে বলা হয় আয়ু টেবিলস্পেস।



চিত্র-১ : আয়ু সেগমেন্ট



চিত্র-২ : আয়ু টেবিলস্পেস গরোজনীয়তা

বুঝা যায়।
ডিকশনারি ম্যানেজড : ডাটা ডিকশনারি দিয়ে এক্সটেন্ড ম্যানেজ হতে থাকে। ওরাকল সার্ভার ডাটা ডিকশনারি গরোজনীয় টেবিল কখন কোথায় স্থান বরাদ্দ হলে বা বরাদ্দ হলো তা নিয়ন্ত্রিত অলাভেই করে থাকে।

লোকালি ম্যানেজড টেবিলস্পেস তৈরির কমান্ড :

আম্বু প্রয়োজনীয়তা (চিত্র-২) : ট্রানজেকশন রোল ব্যাক : আমরা আগেই বলেছি আন্ডু আসলে পুরনো ডাটা সংরক্ষণ করে। যদি ইউজারের ওই ডাটা পুনরুদ্ধার করার দরকার হয়, তখন ওরাকল সার্ভার আন্ডু সেগমেন্ট থেকে আগের প্রকৃত ডাটায় নিয়ে আসে।

ট্রানজেকশন রিকভারি : যদি ট্রানজেকশন কোনো কারণে সম্পূর্ণ না হয় তবে আবার ইনস্ট্যান্স রিগেটন করার সময় আনকমিউটেড পরিবর্তনগুলো রিকভার করার প্রয়োজন হয়। তখন সার্ভার আন্ডু সেগমেন্ট রিড করে ডাটাবেজ ওপেন করে।

রিড কনসিস্ট্যান্সি : যখন কোনো ট্রানজেকশন চলে তখন অন্য কোনো ইউজারের মাধ্যমে পরিবর্তিত আনকমিউটেড ডাটা অন্য সব ইউজারের কোয়রিরেট বাধার সৃষ্টি না করে তার জন্য সার্ভার কোরেট সিস্টেম টেইন নামার নির্ধার করে এবং বড় কোয়ারির রান করার মধ্যেই যদি কোনো পরিবর্তন হয় তাহলেও রিড কনসিস্ট্যান্সি ইমজেক্ট করে ইউজারের প্রয়োজনমতো ফলাফল প্রদর্শন করে।

আন্ডু টেবিলস্পেস তৈরির কমান্ড

```
CREATE UNDO TABLESPACE undo1
DATAFILE 'u01/oradata/undo101.dbf'
SIZE 40M;
```

আন্ডু টেবিলস্পেসে এক্সটেন্ড লোকালি ম্যানুয়াল হয়ে থাকে।

টেম্পোরারি টেবিলস্পেস : যখন কোনো সার্ট অপারেশন ঘটে, তখন মেমরিতে বেশ বড় একটা জায়গার প্রয়োজন হয়। কিন্তু আমরা যদি একটি টেম্পোরারি টেবিলস্পেস তৈরি করি, তাহলে মেমরির ওপর ডান সৃষ্টি না করেই সার্ভার কাজটা সম্পূর্ণ করা যায়। এই টেম্পোরারি টেবিলস্পেসে কোনো স্থায়ী অবজেক্ট রাখা যায় না। এই টেবিলস্পেসের এক্সটেন্ড লোকালি ম্যানুয়াল করতে হয়।

টেম্পোরারি টেবিলস্পেস তৈরির কমান্ড

```
CREATE TEMPORARY TABLESPACE
temp
TEMPFILE 'u01/oradata/temp01.dbf'
SIZE 20M
EXTENT MANAGEMENT LOCAL UNIFORM
SIZE 4M;
```

ডিক্রিট টেম্পোরারি টেবিলস্পেস : যদি কোনো ডাটাবেজের জন্য টেম্পোরারি টেবিলস্পেস নির্ধার না করা হয়, তবে সার্ট অপারেশন সিস্টেম টেবিলস্পেসে হবে। যার ফলে ডাটাবেজের পারফরমেন্স ভালো হয় না। তাই আমাদের অবশ্যই ডাটাবেজের জন্য একটি ডিক্রিট টেম্পোরারি টেবিলস্পেস নির্ধারণ করতে হবে। দু'ভাবে এটা করা যায়।

১. CREATE DATABASE কমান্ড দিয়ে।
২. ALTER DATABASE কমান্ড দিয়ে।
৩. CREATE DATABASE কমান্ড : ডাটাবেজ তৈরির সময় আমরা ডিক্রিট

টেম্পোরারি টেবিলস্পেস নির্ধারণ করতে পারি।

```
CREATE DATABASE DBA01
LOGFILE
GROUP 1 ('$HOME/ORADATA/u01/redo01.log') SIZE 100M,
GROUP 2 ('$HOME/ORADATA/u02/redo02.log') SIZE 100M,
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
MAXINSTANCES 1
DATAFILE '$HOME/ORADATA/u01/system01.dbf' SIZE 325M
UNDO TABLESPACE undo01
DATAFILE '$HOME/ORADATA/u02/undo01.dbf' SIZE 200
DEFAULT TEMPORARY TABLESPACE
temp
TEMPFILE '$HOME/ORADATA/u03/temp01.dbf' SIZE 4M
CHARACTER SET US7ASCII
```

০২. **অস্টার ডাটাবেজ কমান্ড :** ডাটাবেজ তৈরির পর যদি কোনো কারণে আমাদের টেম্পোরারি টেবিলস্পেস নির্ধারণ করতে হয় তবে এই কমান্ড দিয়ে করতে হবে।

```
ALTER DATABASE
DEFAULT TEMPORARY TABLESPACE
default temp2;
```

রিড অনলি টেবিলস্পেস : কোনো কারণে টেবিলস্পেসের কোনো অবজেক্ট যেমন টেবিল, ইনডেক্স ড্রপ করতে হয়, তখন টেবিলস্পেসকে রিড অনলি মোডে নিতে হয়। নিচের কমান্ড দিয়ে টেবিলস্পেসকে রিড অনলি মোডে নিতে হবে।

```
ALTER TABLESPACE userdata READ ONLY;
```

টেবিলস্পেসকে অফলাইনকরণ : সাধারণত টেবিলস্পেস অনলাইন থাকে, তবে নিচে বর্ণিত কারণে কখনো কখনো টেবিলস্পেসকে অফলাইন করা হয়।

০১. ডাটাবেজ পার্টিশন করার সময়।
০২. অফলাইন ব্যাকআপ নেয়ার সময়।
০৩. ডাটাবেজ ওপেন অবস্থায় রিকভারি করলে।
০৪. ডাটা ফাইল মুক্ত করার সময়।

টেবিলস্পেসকে অফলাইনকরণ করার কমান্ড :
ALTER TABLESPACE userdata OFFLINE;

রিসাইজকরণ : দুই উপায়ে টেবিলস্পেস রিসাইজ করা যায়।

০১. ডাটা ফাইলের সাইজ পরিবর্তন করে।
০২. AUTOEXTEND ব্যবহার করে স্বয়ংক্রিয়ভাবে।

```
CREATE TABLESPACE user_data
DATAFILE 'u01/oradata/userdata01.dbf'
SIZE 200M
```

AUTOEXTEND ON NEXT 10M MAX-SIZE 500M;

৪. ম্যানুয়ালি ALTER DATABASE কমান্ড দিয়ে

```
ALTER DATABASE
DATAFILE 'u03/oradata/userdata02.dbf'
RESIZE 200M;
```

০২. ALTER TABLE SPACE কমান্ড দিয়ে কোনো ডাটা ফাইল সংযুক্ত করতে।

```
ALTER TABLESPACE user_data
ADD DATAFILE 'u01/oradata/userdata03.dbf'
SIZE 200M;
```

ডাটা ফাইল স্থানান্তর করা : নিম্নলিখিত ধাপগুলো অনুসরণ করে ডাটা ফাইল স্থানান্তর করা যায়।

০১. টেবিলস্পেস অফলাইন করতে হবে।
০২. অপারেটিং সিস্টেম কমান্ড দিয়ে ফাইলকে মুক্ত করতে হবে।
০৩. নিচের কমান্ড নিতে হবে :

```
ALTER TABLESPACE userdata RENAME
DATAFILE 'u01/oradata/userdata01.dbf'
TO 'u02/oradata/userdata01.dbf';
```

০৪. টেবিলস্পেস অনলাইন করতে হবে।
০৫. প্রয়োজন হলে আগের ফাইল কমান্ড দিয়ে মুক্ত করতে হবে।

TABLE SPACE কে ড্রপ করা :
কোনো অ্যাকটিভ জুথবা সিস্টেম টেবিলস্পেস ছাড়া সব টেবিলস্পেসকে নিচের কমান্ড দিয়ে ড্রপ করা যাবে।

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

টেবিলস্পেস সংরক্ষণ যেকোনো তথ্য আমরা জানতে পারলে নিচের ডাটা ডিকশনারিগুলো ব্যবহার করে :

০১. টেবিলস্পেসের তথ্য
DBA_TABLESPACES
V_TABLESPACE
০২. ডাটা ফাইলের তথ্য
DBA_DATA_FILES
VSDATAFILE
০৩. টেম্পোরারি টেবিলস্পেসের তথ্য
DBA_TEMP_FILES
VSTEMPFILE (সংক্রান্ত)

কিভাবে? Iftekhar@anfobizsol.com



comjagat.com
You are LIVE