

1) Database Replay	2) Partitioning
3) Edition-Based Redefinition	4) Schema Management DDL With the WAIT Option (DDL_LOCK_TIMEOUT) Invisible Indexes Read-Only Tables Finer Grained Dependency Management
5) Patching and Upgrades, RAC One Node, and Clusterware	6) Data Warehousing and OLAP
7) PL/SQL Performance Automatic SQL Tuning	8) PL/SQL Efficient Coding
9) Transaction Management Flashback Data Archive Flashback Transaction LogMiner	10) Security Tablespace Encryption Case Sensitive Passwords
11) SQL Performance Analyzer & Real-Time SQL Monitoring SQL Performance Analyzer	12) Oracle Exadata Simulator
13) SQL Plan Management Adaptive Cursor Sharing AWR Baseline Enhancements	14) Manageability Automatic Memory Management Statistics Collection Enhancements
15) SQL Access Advisor	16) SQL Operations: Pivot and Unpivot
17) Backup and Recovery Data Recovery Advisor RMAN Automated Maintenance Tasks	18) Resiliency Automatic Health Monitor Automatic Diagnostic Repository
19) Automatic Storage Management Automatic Storage Manager (ASM) Enhancements	20) Compression Table Compression Enhancements
21) Caching and Pooling Query Result Cache PL/SQL Function Result Cache Database Resident Connection Pool	22) Data Guard
23) SecureFiles	24) And Don't Forget Install Oracle Database 11g Upgrading to Oracle Database 11g ADDM Enhancements (for RAC) Direct NFS Fine-Grained Access to Network Services Online Table Redefinition Resource Manager Enhancements Scheduler Enhancements Temporary Tablespace Enhancements

1 – DATABASE REPLAY

The Database Replay functionality of Oracle 11g allows you to capture workloads on a production system and replay them exactly as they happened on a test system. This provides an accurate method to test the impact of a variety of system change including:

- Database upgrades
- Operating system upgrades or migrations
- Configuration changes, such as changes to initialization parameters or conversion from a single node to a RAC environment.
- Hardware changes or migration.

2 – PARTITIONING

- Extended Composite Partitioning
- Interval Partitioning
- System Partitioning
- Reference Partitioning
- Virtual Column-Based Partitioning
- Single Partition Transportable for Oracle Data Pump
- Partition Advisor
- Enhanced Statistics Transportable for Oracle Data Pump
- List Partitioning
- Global Index Maintenance
- Hash partitionierte IOTs (Index Organized Tables)

3 – EDITION-BASED REDEFINITION

It's revolutionary: Patch or update your application's data objects while the application remains in uninterrupted use (in Release 2 only)

4 – SCHEMA MANAGEMENT

DDL WITH THE WAIT OPTION (DDL_LOCK_TIMEOUT)

DDL commands require exclusive locks on internal structures. If these locks are not available the commands return with an "ORA-00054: resource busy" error message. This can be especially frustrating when trying to modify objects that are accessed frequently. To get round this Oracle 11g includes the DDL_LOCK_TIMEOUT parameter, which can be set at instance or session level using the ALTER SYSTEM and ALTER SESSION commands respectively.

The DDL_LOCK_TIMEOUT parameter indicates the number of seconds a DDL command should wait for the locks to become available before throwing the resource busy error message. The default value is zero. To see it in action, create a new table and insert a row, but don't commit the insert.

INVISIBLE INDEXES

Oracle 11g allows indexes to be marked as invisible. Invisible indexes are maintained like any other index, but they are ignored by the optimizer unless the OPTIMIZER_USE_INVISIBLE_INDEXES parameter is set to TRUE at the instance or session level. Indexes can be created as invisible by using the INVISIBLE keyword, and their visibility can be toggled using the ALTER INDEX command.

READ-ONLY TABLES

In previous Oracle releases, tables could be made to appear read-only to others users by only granting the SELECT object privilege to them, but the tables remained read-write for the owner. Oracle 11g allows tables to be marked as read-only using the ALTER TABLE command.

```
ALTER TABLE table_name READ ONLY;
```

ALTER TABLE table_name READ WRITE;

FINER GRAINED DEPENDENCY MANAGEMENT

In previous releases, object dependencies were managed at the object level, so altering an object automatically invalidated all dependent objects. Oracle 11g has more granular dependency management, so only changes that directly affect an object will cause an invalidation.

To show this create a table with a dependent package, which is in turn used by a view.

5 - PATCHING AND UPGRADES, RAC One Node, and Clusterware

Enable a single name for the cluster, enable HA for a single-instance database, place OCR and voting disks on ASM

6 – DATA WAREHOUSING AND OLAP

Cube Organized MVS, a new Analytic Workspace Manager, Query Rewrite extended to sub queries and remote tables.

7 - PL/SQL PERFORMANCE

Explore in-lining of code, “real” native compilation, PLS timer, use of simple integer, and more

8 – PL/SQL EFFICIENT CODING

9 – TRANSACTION MANAGEMENT

FLASHBACK DATA ARCHIVE (Oracle Total Recall)

Most flashback features work at the logical level, in that they don’t directly allow you to recover an object. They simply give you a view of how the data was in the past, which in turn can be used to recover from logical corruption, such as accidental deletions. Typically, this view of the past is constructed using undo segments, which are retained for a period of time indicated by the UNDO_RETENTION parameter. Once the undo information is lost, the view of the past is lost also. At least that’s how it was before Oracle 11g introduced the Flashback Data Archive functionality.

A flashback data archive is essentially an extended store of undo information; allowing some logical flashback operations to extend way back into the past with no difference in how they are coded. An individual flashback archive consists of one or more tablespaces, or parts of tablespaces. Each flashback archive has a name, retention period and a quota on each associated tablespace. The database can have multiple flashback data archives, but only a single default archive. When a DML transaction commits an operation on a flashback archive enabled table, the Flashback Data Archiver (FBDA) process stores the pre-image of the rows into a flashback archive, along with metadata of the current rows. The FBDA process is also responsible for managing the data within the flashback archives, such as purging data beyond the retention period.

The following script creates a new tablespace, then creates two flashback data archives using the CREATE FLASHBACK ARCHIVE command. The first is limited in size to 10Gig with a retention period of 1 year, while the second has an unlimited quota and a retention period of 2 years.

FLASHBACK TRANSACTION

As the name implies, Flashback Transaction allows the changes made by a transaction to be undone, optionally including changes made by dependent transactions. The best way to explain this is to see it in action. For this we will be using the new LogMiner screens in Enterprise Manager.

LogMiner requires minimal supplemental logging, but Flashback Transaction requires primary key logging.

LOGMINER

Analyse Interface zu den (archivierten) Redo Log Dateien

In previous versions of Oracle the LogMiner viewer was a separate Java based console, but in Oracle 11g it has been incorporated into Enterprise Manager and integrated with the new Flashback Transaction feature, making it simple to recover transactions that have had an undesirable affect. The logminer functionality is accessed using the “View and Manage Transactions” link on the “Availability” tab.

10 – SECURITY

TABLESPACE ENCRYPTION

The Transparent Data Encryption (TDE) feature was introduced in Oracle 10g Database Release 2 to simplify the encryption of data within datafiles, preventing access to it from the operating system. Tablespace encryption extends this technology, allowing encryption of the entire contents of a tablespace, rather than having to configure encryption on a column-by-column basis.

CASE SENSITIVE PASSWORDS

Case sensitive password (and auditing) are a default feature of newly created Oracle 11g databases.

11- SQL PERFORMANCE ANALYZER & REAL-TIME SQL MONITORING

SQL PERFORMANCE ANALYSER

The concept of SQL tuning sets, along with the DBMS_SQLTUNE package to manipulate them, was introduced in Oracle 10g as part of the Automatic SQL Tuning functionality. Oracle 11g makes further use of SQL tuning sets with the SQL Performance Analyser, which compares the performance of the statements in a tuning set before and after a database change. The database change can be as major or minor as you like, such as:

- Database, operating system, or hardware upgrades.
- Database, operating system, or hardware configuration changes.
- Database initialization parameter changes.
- Schema changes, such as adding indexes or materialized views.
- Refreshing optimizer statistics.
- Creating or changing SQL profiles.

Unlike Database Replay, the SQL Performance Analyzer does not try and replicate the workload on the system. It just Plugs through each statement gathering performance statistics.

The SQL Performance Analyzer can be run manually using the DBMS_SQLPA package or using Enterprise Manager.

12 – ORACLE EXADATA SIMULATOR

13 – SQL PLAN MANAGEMENT

SQL PLAN MANAGEMENT

SQL plan management provides a mechanism for maintaining consistent SQL performance regardless of changes in optimizer version, optimizer statistics, schema changes, system settings and SQL profile creation.

When a SQL statement is hard parsed, the cost based optimizer produces several execution plans and selects the one with the lowest cost. If a SQL plan baseline is present, the optimizer compares the plan it just produced with the plans in the SQL plan baseline. If a matching plan is found that is flagged as accepted the plan is used. If the SQL plan baseline doesn't contain an accepted plan matching the one it just created, the optimizer evaluates the accepted plans in the SQL plan baseline and uses the one with the lowest cost. If the execution plan originally produced by the optimizer has a lower cost than those present in the SQL plan baseline, it is added to the baseline as a not-accepted plan, so it is not used until it is verified not to cause a reduction in performance. If a system change affects all existing accepted plans, so they are considered non-reproducible, the optimizer will use the plan with the lowest cost.

Oracle call this a "conservative plan selection strategy", as the optimizer preferentially uses a tried and tested execution plan, even if a new plan looks like it might perform better. Only when the newer plan is proved to perform well will it be accepted for use.

ADAPTIVE CURSOR SHARING

Erweiterungen: CURSOR Expression

If you want to develop a non-scalable and non-available Oracle application running slowly, then you have one thing to do: "don't use **bind variable**". Oracle architecture is so that sharing memory (**SGA-Library cache**) represents a crucial aspect Oracle engineers have to know and to master. However, as it is always the case with Oracle database, despite this feature is very important it has few drawbacks that are worth to be known. While bind variables allow sharing of **parent cursors** (SQL code) they also allow sharing of **execution plans (child cursor)**. Sharing the same execution plan for different bind variables is not always optimal as far as different bind variables can generate different data volume. This is why Oracle introduces bind **variable peeking** feature which allows Oracle to peek at the bind variable value and give it the best execution plan possible. However, bind variable peeking occurs only at hard parse time which means as far as the query is not hard parsed it will share the same execution plan that corresponds to the last hard parsed bind variable. In order to avoid such situation Oracle introduces in its 11gR2 release, **Adaptive Cursor Sharing** allowing Oracle to adapt itself to the bind variable when necessary without having to wait for a hard parse of the query.

AWR BASELINE ENHANCEMENTS

The Automatic Workload Repository (AWR) was introduced in Oracle 10g and included some simple baseline functionality. Creating a baseline allowed a specified range of snapshots to be retained, regardless of the AWR retention policy, and used for performance comparisons. This functionality, and the DBMS_WORKLOAD_REPOSITORY package that manages it, has been extended in Oracle 11g.

- Fixed Baselines
- The Moving Window Baseline
- Baseline Templates
- Baseline Metric Thresholds

14 - MANAGEABILITY

AUTOMATIC MEMORY MANAGEMENT

Dynamisches Memory Management

Erlaubt dynamische Änderungen an den wichtigsten Caches (SGA) und ebenfalls automatisches „SQL Execution Memory Management“ für die PGAs

Oracle has made great strides in simplifying memory management over the last few versions of the database. Oracle 9i automated PGA management by introducing PGA_AGGREGATE_TARGET parameter. Oracle 10g continued this

trend by automating SGA management using SGA_TARGET parameter. Oracle 11g takes this one step further by allowing you to allocate one chunk of memory, which Oracle uses to dynamically manage both the SGA and PGA.

We can expand or shrink both SGA and PGA based on the instance demands.

MEMORY_TARGET: This parameter sets the system-wide usable memory that will be used by the instance for SGA and PGA.

MEMORY_MAX_TARGET: This defines the maximum size the MEMORY_TARGET can be increased to without an instance restart. If the MEMORY_MAX_TARGET is not specified, it defaults to MEMORY_TARGET setting.

Enabling automatic memory management (AMM):

On linux system make sure you have configure /dev/shm before enabling automatic memory management.

/dev/shm is nothing but implementation of traditional **shared memory** concept. It is an efficient means of passing data between programs.

The shared memory file system should be big enough to accommodate the MEMORY_TARGET and MEMORY_MAX_TARGET values, or Oracle will throw the following error:

ORA-00845: MEMORY_TARGET not supported on this system

STATISTICS COLLECTION ENHANCEMENTS

- Pending Statistics
- Extended Statistics
 - Multi-Column Statistics
 - Expression Statistics
- Enhanced Statistics Collection for Partitioned Objects

15 – SQL ACCESS ADVISOR

The SQL Access Advisor was introduced in Oracle 10g to make suggestions about additional indexes and materialized Views which might improve system performance. Oracle 11g has made two significant changes to the SQL Access Advisor:

1. The advisor now includes advice on partitioning schemes that may improve performance.
2. The original workload manipulation has been deprecated and replaced by SQL tuning sets.

16 – SQL OPERATIONS: PIVOT AND UNPIVOT

17 – BACKUP AND RECOVERY

DATA RECOVERY ADVISOR

The Data Recovery Advisor automatically diagnoses corruption or loss of persistent data on disk, determines the appropriate repair options, and executes repairs at the user's request. This reduces the complexity of recovery process, thereby reducing the Mean Time To Recover (MTTR).

RMAN> LIST FAILURE;

RMAN> ADVISE FAILURE;

RMAN> REPAIR FAILURE PREVIEW;

RMAN> REPAIR FAILURE NOPROMPT;

RMAN

Improved Integration with Data Guard. When RMAN is in a Data Guard environment, the CONFIGURE command allows you to register and configure settings for all physical databases. A catalog must be connected for RMAN to work properly in a Data Guard environment.

AUTOMATED MAINTENANCE TASKS

Oracle 11g includes three automated database maintenance tasks:

- Automatic Optimizer Statistics Collection – Gathers stale or missing statistics for all schema objects.
(Task name: auto optimizer stats collection)
- Automatic Segment Advisor – Identifies segments that could be reorganized to save space.
(Task name: auto space advisor)
- Automatic SQL Tuning Advisor – Identifies and attempts to tune high load SQL.
(Task name: SQL tuning advisor)

18 - RESILIENCY

HEALTH MONITOR (DBMS_HM)

In the previous example the failure was detected and logged automatically in reaction to the table creation failure.

If this were a low use tablespace, we might not have noticed the problem for some time. The Health Monitor also allows us to perform the same integrity checks manually, rather than waiting for the reactive tests to take place. This may help you identify and fix problems before they are ever noticed by users.

Access to Health Monitor is available using the DBMS_HM package or Enterprise Manager (see below). The available checks are displayed using the V\$HM_CHECK view.

DIAGNOSTICS REPOSITORY

Managing Diagnostic Data

Beginning with Release 11g, Oracle Database includes an advanced fault diagnosability infrastructure for collecting and managing diagnostics data. Diagnostic data includes the trace files, dumps, and core files that are also present in previous releases, plus new types of diagnostic data that enable customers and Oracle Support to identify, investigate, track, and resolve problems quickly and effectively.

- About the Oracle Database Fault Diagnosability Infrastructure
- Investigating, Reporting, and Resolving a Problem
- Viewing Problems with the Enterprise Manager Support Workbench
- Creating a User-Reported Problem
- Viewing the Alert Log
- Finding Trace Files
- Running Health Checks with Health Monitor
- Repairing SQL Failures with the SQL Repair Advisor
- Repairing Data Corruptions with the Data Recovery Advisor
- Creating, Editing, and Uploading Custom Incident Packages

19 – AUTOMATIC STORAGE MANAGEMENT

20 - COMPRESSION

TABLE COMPRESSION ENHANCEMENTS

Table compression was introduced in Oracle 9i as a space saving feature for data warehousing projects. In 11g it is now considered a mainstream feature that is acceptable for OLTP databases. In addition to saving storage space, compression can result in increased I/O performance and reduced memory use in the buffer cache. These advantages do come at a cost, since compression incurs a CPU overhead, so it won't be of benefit to everyone.

The compression clause can be specified at the tablespace, table or partition level with the following options:

- **NOCOMPRESS** : The table or partition is not compressed. This is the default action when no compression clause is specified.
- **COMPRESS** : This option is considered suitable for data warehouse systems. Compression is enabled on the table or partition during direct-path inserts only.
- **COMPRESS FOR DIRECT_LOAD OPERATIONS**: This option has the same effect as the simple **COMPRESS** keyword.
- **COMPRESS FOR ALL OPERATIONS**: This option is considered suitable for OLTP systems. As the name implies, this option enables compression for all operations, including regular DML statements. This option requires the **COMPATIBLE** initialization parameter to be set to 11.1.0 or higher. In 11gR2 this option has been renamed to **COMPRESS FOR OLTP** and the original name has been deprecated.

21 – CACHING AND POOLING

QUERY RESULT CACHE

Oracle 11g allows the results of SQL queries to be cached in the SGA and reused to improve performance.

FUNCTION RESULT CACHE

The cross-session PL/SQL function result cache provides a simple way to boost the performance of PL/SQL functions by saving the results of function calls for specific combinations of input parameters in the SGA. These results can be reused by any session calling the same function with the same parameters. This can be a significant performance boost when functions are called for each row in an SQL query, or with a loop in PL/SQL.

DATABASE RESIDENT CONNECTION POOL

The database resident connection pool (DRCP) reduces the resource requirements of applications that currently don't support connection pooling, either because it is not supported by the application infrastructure, or it has not been implemented. DRCP is only supported for database connections using the OCI driver.

DBMS_CONNECTION_POOL

Pooled Connections

Views

DRCP and Java

22 – DATA GUARD

23 - SECUREFILES

The SecureFiles functionality is a complete redesign of the implementation of large object (LOB) storage in Oracle 11g. The original LOB storage, now known as BASICFILE, is still the default storage method, but the SECUREFILE keyword enables the new storage method, which allows encryption for security and space savings using compression and deduplication.

24 – AND DON'T FORGET

ADDM ENHANCEMENTS FOR RAC

ADDM (Automated Database Diagnostic Monitor) examines and analyzes data captured in the Automatic Workload Repository (AWR) to determine possible performance problems in Oracle Database. ADDM then locates the root causes of the performance problems, provides recommendations for correcting them, and quantifies the expected benefits. ADDM analyzes the AWR data for performance problems at both the database and the instance level.

An ADDM analysis is performed as each AWR snapshot is generated, which is every hour by default. The results are saved in the database and can be viewed by using Enterprise Manager. Any time you have a performance problem, you should first review the results of the ADDM analysis. An ADDM analysis is performed from the top down, first identifying symptoms, then refining the analysis to reach the root causes, and finally providing remedies for the problems.

DIRECT NFS Client

For improved NFS performance, Oracle recommend using the Direct NFS Client shipped with Oracle 11g. The direct NFS client looks for NFS detail in the following locations.

1. \$ORACLE_HOME/dbs/oranfstab
2. /etc/oranfstab
3. /etc/mtab

FINE GRAINED ACCESS TO NETWORK SERVICES

Oracle allows access to external network services using several PL/SQL APIs (UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP and UTL_INADDR), all of which are implemented using the TCP protocol. In previous versions of the database, access to external services was effectively an on/off switch based on whether a user was granted execute permissions on a specific package or not. Oracle 11g introduces fine grained access to network services using access control lists (ACL) in the XML DB repository, allowing control over which users access which network resources, regardless of package grants.

INSTALL ORACLE DATABASE 11G

ONLINE TABLE REDEFINITION

By default, online table redefinitions no longer invalidate dependent objects (PL/SQL, views, synonyms etc.), provided the redefinition does not logically affect them. An exception to this behavior is triggers, which are associated directly with a table.

Tabellen Reorganisationen nun auch online möglich (7 x 24)

- Voller DML-Support während Reorg
- Table zweimal nur während kurzer Zeit gelockt

RESOURCE MANAGER ENHANCEMENTS

Overview of all the resource manager enhancements in the Enterprise Edition of Oracle Database 11g Release 1, including the following.

- I/O Calibration
- Per Session I/O Limits

- Resource Manager Statistics in AWR
- Built-In Resource Plans

Mit Oracle8i eingeführt, um Ressourcen gezielt an User zuordnen zu können

- Prozentuale CPU-Limite
- Maximale Parallelisierungsgrad

SCHEDULER ENHANCEMENTS

Oracle 10g Release 1 introduced the new Oracle scheduler and Oracle 10g Release 2 enhanced it. Now Oracle 11g adds more functionality to the Oracle scheduler.

- Remote External Jobs
 - o Database Configuration
 - o Oracle Scheduler Agent Installation
 - o Credentials
 - o Creating Remote External Jobs
 - o Retourning stdout and stderr
 - o Disabling Remote External Job Functionality
- Detached Jobs
- Lightweight Jobs
- Scheduler Support for Data Guard
- Oracle Enterprise Manager

SQL ACCESS ADVISOR

The SQL Access Advisor was introduced in Oracle 10g to make suggestions about additional indexes and materialized views which might improve system performance. Oracle 11g has made two significant changes to the SQL Access Advisor.

1. The advisor now includes advice on partitioning schemes that may improve performance.
2. The original workload manipulation has been deprecated and replaced by SQL tuning sets.

TEMPORARY TABLESPACE ENHANCEMENTS

Es kann ein Default Temporary Tablespace für alle User beim create database angegeben werden (locally managed)

Oracle 11g has a new view called DBA_TEMP_FREE_SPACE that displays information about temporary tablespace usage.

```
SQL> SELECT * FROM dba_temp_free_space;
```

```
SQL> ALTER TABLESPACE temp SHRINK SPACE KEEP 40M;
```

```
SQL> ALTER TABLESPACE temp SHRINK TEMPFILE '/u01/app/oracle/oradata/DB11G/temp01.dbf' KEEP 30M;
```

UPGRADING TO ORACLE DATABASE 11G