Difference between Oracle 10g and 11g

1- **Simplified and improved automatic memory management**

Oracle 9i automated PGA management by introducing `PGA_AGGREGATE_TARGET` parameter. Oracle 10g continued this trend by automating SGA management using the `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.

- **MEMORY_TARGET**: The amount of shared memory available for Oracle to use when dynamically controlling the SGA and PGA. This parameter is dynamic, so the total amount of memory available to Oracle can be increased or decreased, provided it does not exceed the `MEMORY_MAX_TARGET` limit. The default value is "0".

- **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.

2- **New fault diagnosability infrastructure to prevent, detect, diagnose, and help resolve critical database errors.**

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error (such as trace files) are immediately captured and tagged with this number. The data is then stored in the **Automatic Diagnostic Repository (ADR)**—a file-based repository outside the database—where it can later be retrieved by incident number and analyzed.

The goals of the fault diagnosability infrastructure are the following:

- First-failure diagnosis
- Problem prevention
- Limiting damage and interruptions after a problem is detected
- Reducing problem diagnostic time
- Reducing problem resolution time
- Simplifying customer interaction with Oracle Support

```
SELECT * FROM V$DIAG_INFO;
```

<table>
<thead>
<tr>
<th>INST_ID</th>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Diag Enabled</td>
<td>TRUE</td>
</tr>
<tr>
<td>1</td>
<td>ADR Base</td>
<td>/u01/oracle</td>
</tr>
<tr>
<td>1</td>
<td>ADR Home</td>
<td>/u01/oracle/diag/rdbms/orcl/orci</td>
</tr>
<tr>
<td>1</td>
<td>Diag Trace</td>
<td>/u01/oracle/diag/rdbms/orcl/orci/trace</td>
</tr>
<tr>
<td>1</td>
<td>Diag Alert</td>
<td>/u01/oracle/diag/rdbms/orcl/orci/alert</td>
</tr>
<tr>
<td>1</td>
<td>Diag Incident</td>
<td>/u01/oracle/diag/rdbms/orcl/orc/incident</td>
</tr>
<tr>
<td>1</td>
<td>Diag Cdump</td>
<td>/u01/oracle/diag/rdbms/orcl/orcl/cdump</td>
</tr>
<tr>
<td>1</td>
<td>Health Monitor</td>
<td>/u01/oracle/diag/rdbms/orcl/orcl/hm</td>
</tr>
<tr>
<td>1</td>
<td>Default Trace File</td>
<td>/u01/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_22769.trc</td>
</tr>
<tr>
<td>1</td>
<td>Active Problem Count</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>Active Incident Count</td>
<td>20</td>
</tr>
</tbody>
</table>

To obtain a list of health check names, run the following query:

```
SELECT name FROM v$hm_check WHERE internal_check='N';
```
DB Structure Integrity Check
Data Block Integrity Check
Redo Integrity Check
Transaction Integrity Check
Undo Segment Integrity Check
Dictionary Integrity Check

To run a Health Monitor Checker using Enterprise Manager

- On the Database Home page, in the Related Links section, click Advisor Central.
- Click Checkers to view the Checkers subpage
- In the Checkers section, click the checker you want to run.
- Enter values for input parameters or, for optional parameters, leave them blank to accept the defaults.
- Click Run, confirm your parameters, and click Run again.

3- Invisible Indexes

If the usage of the index is at all, or it results a bad performance to other queries, one can easily drop the index. However in future, if some adhoc queries require the index badly (where the index dropped would have increased the performance dramatically), we need to think reinstating the index again. The recreation of an index is a very expensive one and requires a lot of database resources and consumes a lot of time as well.

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.

CREATE INDEX index_name ON table_name(column_name) INVISIBLE;
ALTER INDEX index_name INVISIBLE;
ALTER INDEX index_name VISIBLE;

SELECT * FROM EMPLOYEES WHERE EMPID = 1001; (Full table scanning)
Adhoc queries requiring the index using index hint:
SELECT /*+ index(index_name) */ * FROM EMPLOYEES WHERE EMPID = 1001;

4- Virtual columns

When queried, virtual columns appear to be normal table columns, but their values are derived rather than being stored on disc.

The syntax for defining a virtual column is listed below.

```sql
column_name [datatype] [GENERATED ALWAYS] AS (expression) [VIRTUAL]
```
If the data type is omitted, it is determined based on the result of the expression. The GENERATED ALWAYS and VIRTUAL keywords are provided for clarity only.

The script below creates and populates an employees table with two levels of commission. It includes two virtual columns to display the commission-based salary. The first uses the most abbreviated syntax while the second uses the most verbose form.

```sql
CREATE TABLE employees (
    id          NUMBER,
    first_name  VARCHAR2(10),
    last_name   VARCHAR2(10),
    salary      NUMBER(9,2),
    comm1       NUMBER(3),
    comm2       NUMBER(3),
    salary1     AS (ROUND(salary*(1+comm1/100),2)) VIRTUAL,
    salary2     NUMBER GENERATED ALWAYS AS (ROUND(salary*(1+comm2/100),2)) VIRTUAL,
    CONSTRAINT employees_pk PRIMARY KEY (id)
);
```

Moreover, virtual columns can be referenced in the WHERE clause of updates and deletes, but they cannot be manipulated by DML.

In my opinion it’s the optimizer statistics that are likely to be the most common benefit of virtual columns.

5- **Enhanced security for password-based authentication by enabling use of mixed case in passwords.**

Passwords are case sensitive in Oracle 11g, while user names are still case insensitive as before. To support mixed case password names, the orapwd program has been modified to allow or disallow case-sensitive passwords and the ignorecase parameter has been added to the command line. Ignorecase should be set to Y to enable password case sensitivity for SYS and SYSDBA connections.

```sql
ALTER SYSTEM set sec_case_sensitive_logon = false;
ALTER SYSTEM set sec_case_sensitive_logon = true;
```

6- **Tablespace-level 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 tablescape, rather than having to configure encryption on a column-by-column basis.