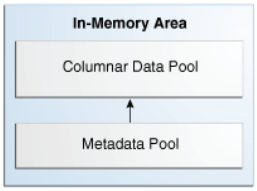
Figure 2-3 Subpools in the In-Memory Area



Description of "Figure 2-3 Subpools in the In-Memory Area"

The database determines the relative size of the two subpools using internal heuristics. The database allocates t of space in the In-Memory Area to the columnar data pool (1 MB pool).

Note: Oracle Database automatically determines the subpool sizes. You cannot change the space allocations.

Example 2-1 V$INMEMORY_AREA View

This example queries the `V$INMEMORY_AREA` view to determine the amount of available memory in each subpo output included):

```
COL POOL FORMAT a9
COL POPULATE_STATUS FORMAT a15
SSELECT POOL, TRUNC(ALLOC_BYTES/(1024*1024*1024),2) "ALLOC_GB",
        TRUNC(USED_BYTES/(1024*1024*1024),2) "USED_GB",
        POPULATE_STATUS
FROM    V$INMEMORY_AREA;

POOL      ALLOC_GB   USED_GB    POPULATE_STATUS
--------- ---------- ---------- ---------------
1MB POOL  7.99       0          DONE
64KB POOL 1.98       0          DONE
```

The current size of the In-Memory area is visible in `V$SGA`:

```
SELECT NAME, VALUE/(1024*1024*1024) "SIZE_IN_GB"
FROM   V$SGA
WHERE  NAME LIKE '%Mem%';

NAME                SIZE_IN_GB
------------------- ----------
In-Memory Area      10
```

In this example, the memory allocated to the subpools is 9.97 GB, whereas the size of the In-Memory Area is 10 database uses a small percentage of memory for internal management structures.

See Also:

Oracle Database Reference to learn about `V$INMEMORY_AREA`

2.1.2 Row Data in the Database Buffer Cache

The database buffer cache stores and processes data blocks in the same way whether the IM column store is ena disabled. Buffer I/O and buffer pools function the same.

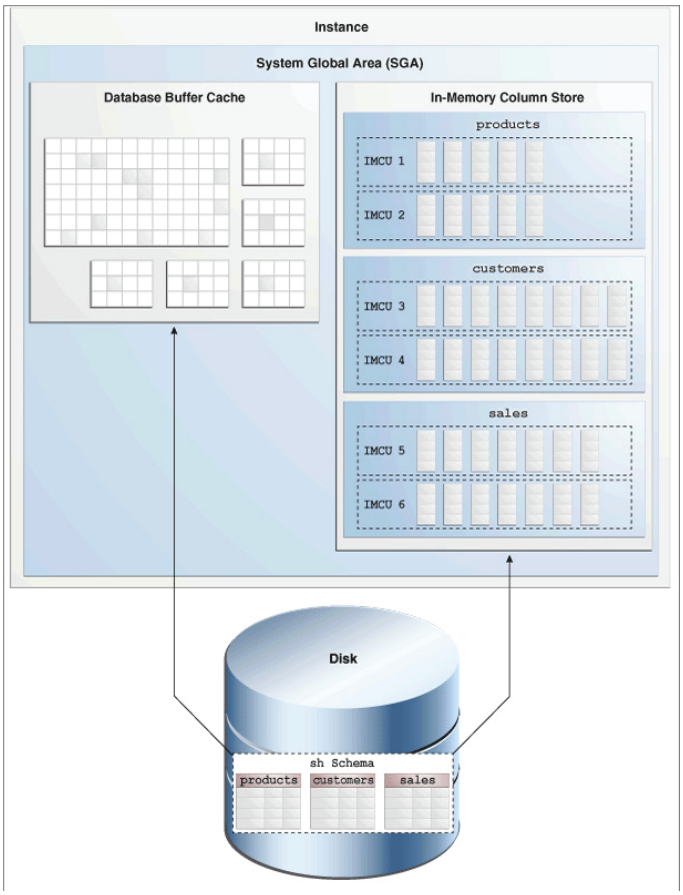
The IM column store enables data to be simultaneously populated in the SGA in both the traditional row format (cache) and the columnar format. The database transparently sends OLTP queries (such as primary key lookups) buffer cache, and analytic and reporting queries to the IM column store. When fetching data, Oracle Database ca data from both memory areas within the same query.

Note: In the execution plan, the operation `TABLE ACCESS IN MEMORY FULL` indicates that some data is accessed in the IM column store.

The dual-format architecture does not double memory requirements. The buffer cache is optimized to run with a smaller size than the size of the database.

The following figure shows a sample IM column store. The database stores the `sh.sales` table on disk in traditi format. The SGA stores the data in columnar format in the IM column store, and in row format in the database bu

Figure 2-4 IM Column Store



Description of "Figure 2-4 IM Column Store"

44

Every on-disk data format for permanent, heap-organized tables is supported by the IM column store. The colun does not affect the format of data stored in data files or in the buffer cache, nor does it affect undo data and onli logging.

The database processes DML modifications in the same way, regardless of whether the IM column store is enabl updating the buffer cache, online redo log, and undo tablespace. However, the database uses an internal mecha track changes and ensure that the IM column store is consistent with the rest of the database. For example, if the table is populated in the IM column store, and if an application updates a row in `sales`, then the database auton keeps the copy of the `sales` table in the IM column store transactionally consistent. A query that accesses the IM store always returns the same results for a query that accesses the buffer cache.

See Also:

Oracle Database Concepts to learn more about the database buffer cache

2.2 In-Memory Storage Units

The IM column store manages both data and metadata in optimized storage units, not in traditional Oracle data

Oracle Database maintains the storage units in the In-Memory Area. The following graphic gives an overview of Memory Area and the database processes that interact with it. The remaining chapter describes the various mem components.

Figure 2-5 IM Column Store: Memory and Process Architecture