

U1M9.LW.Partitioning

Shkrabatouskaya Vera

https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022

2. Oracle Architecture - Partitioning

2.1. Task 01:

CREATE Example of Range partitioning

Example of range partitioning table. Perform Administration tasks on all partitioning types:

Maintenance Operation	Range Composite Range-*	Interval Composite Interval-*	Hash	List Composite List-*	Reference
Adding Partitions	ADD PARTITION	ADD PARTITION (If data is loaded into this table with a later date than the greatest defined partition, Oracle will automatically create a new partition for the new month.)	ADD PARTITION	ADD PARTITION	N/A Foot 1
Coalescing Partitions	N/A	N/A	COALESCE PARTITION	N/A	N/A Footref 1
Dropping Partitions	DROP PARTITION	DROP PARTITION	N/A	DROP PARTITION	N/A Footref 1
Merging Partitions	MERGE PARTITIONS	MERGE PARTITIONS	N/A	MERGE PARTITIONS	N/A Footref 1
Moving Partitions	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION
Splitting Partitions	SPLIT PARTITION	SPLIT PARTITION	N/A	SPLIT PARTITION	N/A Footref 1
Truncating Partitions	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION	TRUNCATE PARTITION

Range partitions.

Create a calendar table with Range partitions.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, there are tabs for 'Welcome Page', 'VeraDB', 'Lab10_Task1.sql', and 'Lab9_Task1.sql'. The 'Lab10_Task1.sql' tab is active. The main area is a 'Worksheet' tab showing the following SQL code:

```
    (
    PARTITION PART_1 VALUES LESS THAN (TO_DATE('01/01/2013', 'DD/MM/YYYY')),
    PARTITION PART_2 VALUES LESS THAN (TO_DATE('01/01/2014', 'DD/MM/YYYY'))
)
TABLESPACE ts_dw_data_02;
```

Below the worksheet, in the 'Script Output' tab, it says 'Task completed in 0.053 seconds'. The output shows:

```
Session altered.

Table CALENDAR created.
```

Insert values into the table.

The screenshot shows the Oracle SQL Developer interface with the 'Lab10_Task1.sql' tab active. The 'Worksheet' tab contains a complex PL/SQL script for generating a calendar table. The 'Script Output' tab at the bottom shows the results of the execution:

```
Table CALENDAR created.

1,000 rows inserted.
```

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM calendar PARTITION (PART_1);
```

Script Output | Query Result | Fetched 100 rows in 0.025 seconds

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	EP
1 01-JAN-12	Sunday	7	01	001	1	01-JAN-12	01	31	31-
2 02-JAN-12	Monday	1	02	002	1	08-JAN-12	01	31	31-
3 03-JAN-12	Tuesday	2	03	003	1	08-JAN-12	01	31	31-
4 04-JAN-12	Wednesday	3	04	004	1	08-JAN-12	01	31	31-
5 05-JAN-12	Thursday	4	05	005	1	08-JAN-12	01	31	31-
6 06-JAN-12	Friday	5	06	006	1	08-JAN-12	01	31	31-
7 07-JAN-12	Saturday	6	07	007	1	08-JAN-12	01	31	31-
8 08-JAN-12	Sunday	7	08	008	2	08-JAN-12	01	31	31-
9 09-JAN-12	Monday	1	09	009	2	15-JAN-12	01	31	31-
10 10-JAN-12	Tuesday	2	10	010	2	15-JAN-12	01	31	31-
11 11-JAN-12	Wednesday	3	11	011	2	15-JAN-12	01	31	31-
12 12-JAN-12	Thursday	4	12	012	2	15-JAN-12	01	31	31-
13 13-JAN-12	Friday	5	13	013	2	15-JAN-12	01	31	31-
14 14-JAN-12	Saturday	6	14	014	2	15-JAN-12	01	31	31-
15 15-JAN-12	Sunday	7	15	015	3	15-JAN-12	01	31	31-
16 16-JAN-12	Monday	1	16	016	3	22-JAN-12	01	31	31-
17 17-JAN-12	Tuesday	2	17	017	3	22-JAN-12	01	31	31-
18 18-JAN-12	Wednesday	3	18	018	3	22-JAN-12	01	31	31-
19 19-JAN-12	Thursday	4	19	019	3	22-JAN-12	01	31	31-
20 20-JAN-12	Friday	5	20	020	3	22-JAN-12	01	31	31-
21 21-JAN-12	Saturday	6	21	021	3	22-JAN-12	01	31	31-
22 22-JAN-12	Sunday	7	22	022	4	22-JAN-12	01	31	31-
23 23-JAN-12	Monday	1	23	023	4	29-JAN-12	01	31	31-
24 24-JAN-12	Tuesday	2	24	024	4	29-JAN-12	01	31	31-
25 25-JAN-12	Wednesday	3	25	025	4	29-JAN-12	01	31	31-

SQL Worksheet | History

Worksheet | Query Builder

```
SELECT * FROM calendar PARTITION (PART_2);
```

Script Output | Query Result | Fetched 50 rows in 0.016 seconds

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	EP
1 01-JAN-13	Tuesday	2	01	001	1	06-JAN-13	01	31	31-
2 02-JAN-13	Wednesday	3	02	002	1	06-JAN-13	01	31	31-
3 03-JAN-13	Thursday	4	03	003	1	06-JAN-13	01	31	31-
4 04-JAN-13	Friday	5	04	004	1	06-JAN-13	01	31	31-
5 05-JAN-13	Saturday	6	05	005	1	06-JAN-13	01	31	31-
6 06-JAN-13	Sunday	7	06	006	1	06-JAN-13	01	31	31-
7 07-JAN-13	Monday	1	07	007	1	13-JAN-13	01	31	31-
8 08-JAN-13	Tuesday	2	08	008	2	13-JAN-13	01	31	31-
9 09-JAN-13	Wednesday	3	09	009	2	13-JAN-13	01	31	31-
10 10-JAN-13	Thursday	4	10	010	2	13-JAN-13	01	31	31-
11 11-JAN-13	Friday	5	11	011	2	13-JAN-13	01	31	31-
12 12-JAN-13	Saturday	6	12	012	2	13-JAN-13	01	31	31-
13 13-JAN-13	Sunday	7	13	013	2	13-JAN-13	01	31	31-
14 14-JAN-13	Monday	1	14	014	2	20-JAN-13	01	31	31-
15 15-JAN-13	Tuesday	2	15	015	3	20-JAN-13	01	31	31-
16 16-JAN-13	Wednesday	3	16	016	3	20-JAN-13	01	31	31-
17 17-JAN-13	Thursday	4	17	017	3	20-JAN-13	01	31	31-
18 18-JAN-13	Friday	5	18	018	3	20-JAN-13	01	31	31-
19 19-JAN-13	Saturday	6	19	019	3	20-JAN-13	01	31	31-
20 20-JAN-13	Sunday	7	20	020	3	20-JAN-13	01	31	31-
21 21-JAN-13	Monday	1	21	021	3	27-JAN-13	01	31	31-
22 22-JAN-13	Tuesday	2	22	022	4	27-JAN-13	01	31	31-
23 23-JAN-13	Wednesday	3	23	023	4	27-JAN-13	01	31	31-
24 24-JAN-13	Thursday	4	24	024	4	27-JAN-13	01	31	31-
25 25-JAN-13	Friday	5	25	025	4	27-JAN-13	01	31	31-

Merge partitions.

The screenshot shows a SQL Worksheet window with the following content:

```
132 --Merge partitions
133
134 ALTER TABLE calendar
135 MERGE PARTITIONS PART_1, PART_2 INTO PARTITION PART_3;
136
137
```

Below the worksheet, a Script Output window displays the message: "Task completed in 0.187 seconds".

Text output below the windows states: "Table CALENDAR altered."

The screenshot shows a SQL Worksheet window with the following content:

```
133
134 ALTER TABLE calendar
135 MERGE PARTITIONS PART_1, PART_2 INTO PARTITION PART_3;
136
137 SELECT * FROM calendar PARTITION (PART_3);
```

Below the worksheet, a Query Result window displays the following table:

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_DATE
358	23-DEC-12 Sunday	7	23	358	4	23-DEC-12	12	31	31-12-2012
359	24-DEC-12 Monday	1	24	359	4	30-DEC-12	12	31	31-12-2012
360	25-DEC-12 Tuesday	2	25	360	4	30-DEC-12	12	31	31-12-2012
361	26-DEC-12 Wednesday	3	26	361	4	30-DEC-12	12	31	31-12-2012
362	27-DEC-12 Thursday	4	27	362	4	30-DEC-12	12	31	31-12-2012
363	28-DEC-12 Friday	5	28	363	4	30-DEC-12	12	31	31-12-2012
364	29-DEC-12 Saturday	6	29	364	5	30-DEC-12	12	31	31-12-2012
365	30-DEC-12 Sunday	7	30	365	5	30-DEC-12	12	31	31-12-2012
366	31-DEC-12 Monday	1	31	366	5	06-JAN-13	12	31	31-12-2012
367	01-JAN-13 Tuesday	2	01	001	1	06-JAN-13	01	31	31-12-2012
368	02-JAN-13 Wednesday	3	02	002	1	06-JAN-13	01	31	31-12-2012
369	03-JAN-13 Thursday	4	03	003	1	06-JAN-13	01	31	31-12-2012
370	04-JAN-13 Friday	5	04	004	1	06-JAN-13	01	31	31-12-2012
371	05-JAN-13 Saturday	6	05	005	1	06-JAN-13	01	31	31-12-2012
372	06-JAN-13 Sunday	7	06	006	1	06-JAN-13	01	31	31-12-2012
373	07-JAN-13 Monday	1	07	007	1	13-JAN-13	01	31	31-12-2012
374	08-JAN-13 Tuesday	2	08	008	2	13-JAN-13	01	31	31-12-2012
375	09-JAN-13 Wednesday	3	09	009	2	13-JAN-13	01	31	31-12-2012
376	10-JAN-13 Thursday	4	10	010	2	13-JAN-13	01	31	31-12-2012
377	11-JAN-13 Friday	5	11	011	2	13-JAN-13	01	31	31-12-2012
378	12-JAN-13 Saturday	6	12	012	2	13-JAN-13	01	31	31-12-2012

Move partition.

Grant the user u_dw_data_02 the appropriate system privileges or grant the user space resource on the tablespace.

GRANT UNLIMITED TABLESPACE TO u_dw_data_02;

Grant succeeded.

Error starting at line : 161 in command -
ALTER TABLE calendar MOVE PARTITION PART_3 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS
Error report -
ORA-01950: no privileges on tablespace 'TS_DW_DATA_03'
01950. 00000 - "no privileges on tablespace '%s'"
*Cause: User does not have privileges to allocate an extent in the
specified tablespace.
*Action: Grant the user the appropriate system privileges or grant the user
space resource on the tablespace.

Grant succeeded.

Move PARTITION (PART_3) from tablespace ts_dw_data_02 to tablespace ts_dw_data_03.

ALTER TABLE calendar MOVE PARTITION PART_3 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS;

Grant succeeded.

Table CALENDAR altered.

PARTITION (PART_3) is moved.

163 | SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
164 |

Script Output x Query Result x
SQL | All Rows Fetched: 579 in 0.145 seconds

PARTITION_NAME	TABLESPACE_NAME
56/ SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1028	SYSAUX
577 SYS_P1031	SYSAUX
578 PART_3	TS_DW_DATA_03
579 P0	SYSAUX

Split partition.

The screenshot shows a SQL developer interface. In the top window, a script is being run:

```
--Split partition
ALTER TABLE calendar SPLIT PARTITION PART_3
AT (TO_DATE('01/01/2014', 'DD/MM/YYYY'))
INTO
( PARTITION PART_1, PARTITION PART_2);
```

In the bottom window, the output of the command is displayed:

Table CALENDAR altered.

PARTITION (PART_3) is split into PART_1 and PART_2.

The screenshot shows a SQL developer interface with a query result window open. The query executed is:

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The results show the following partitions and their tablespaces:

partition_name	tablespace_name
558 SYS_P875	SYSAUX
559 SYS_P956	SYSAUX
560 SYS_P939	SYSAUX
561 SYS_P942	SYSAUX
562 SYS_P896	SYSAUX
563 SYS_P899	SYSAUX
564 SYS_P979	SYSAUX
565 SYS_P1003	SYSAUX
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 PART_2	TS_DW_DATA_03
577 SYS_P1028	SYSAUX
578 PART_1	TS_DW_DATA_03
579 SYS_P1031	SYSAUX
580 P0	SYSAUX

SELECT * FROM calendar PARTITION (PART_1);											
Script Output		Query Result									
TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	ET		
361	26-DEC-12	Wednesday	3	26	361	4	30-DEC-12	12	31	31-	
362	27-DEC-12	Thursday	4	27	362	4	30-DEC-12	12	31	31-	
363	28-DEC-12	Friday	5	28	363	4	30-DEC-12	12	31	31-	
364	29-DEC-12	Saturday	6	29	364	5	30-DEC-12	12	31	31-	
365	30-DEC-12	Sunday	7	30	365	5	30-DEC-12	12	31	31-	
366	31-DEC-12	Monday	1	31	366	5	06-JAN-13	12	31	31-	
367	01-JAN-13	Tuesday	2	01	001	1	06-JAN-13	01	31	31-	
368	02-JAN-13	Wednesday	3	02	002	1	06-JAN-13	01	31	31-	
369	03-JAN-13	Thursday	4	03	003	1	06-JAN-13	01	31	31-	

Script Output x Query Result x											
1 01-JAN-14 Wednesday	3	01	001	1	05-JAN-14	01	31	31-			
2 02-JAN-14 Thursday	4	02	002	1	05-JAN-14	01	31	31-			
3 03-JAN-14 Friday	5	03	003	1	05-JAN-14	01	31	31-			
4 04-JAN-14 Saturday	6	04	004	1	05-JAN-14	01	31	31-			
5 05-JAN-14 Sunday	7	05	005	1	05-JAN-14	01	31	31-			
6 06-JAN-14 Monday	1	06	006	1	12-JAN-14	01	31	31-			
7 07-JAN-14 Tuesday	2	07	007	1	12-JAN-14	01	31	31-			
8 08-JAN-14 Wednesday	3	08	008	2	12-JAN-14	01	31	31-			
9 09-JAN-14 Thursday	4	09	009	2	12-JAN-14	01	31	31-			

Truncate partition.

```
1 / 3  
174 --Truncate partition  
175 ALTER TABLE calendar TRUNCATE PARTITION PART_2;  
  
Script Output X  
| Task completed in 0.038 seconds  
  
Table CALENDAR truncated.
```

PARTITION (PART_2) is truncated.

Drop partition.

```
177 | --Drop partition
178 | ALTER TABLE calendar DROP PARTITION PART_2;
```

Script Output X | Query Result X
Task completed in 0.032 seconds

Table CALENDAR truncated.

Table CALENDAR altered.

PARTITION (PART_2) is dropped.

```
181 |
182 | SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

Script Output X | Query Result X
SQL | All Rows Fetched: 579 in 0.117 seconds

PARTITION_NAME	TABLESPACE_NAME
570 PART_004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1028	SYSAUX
577 PART_1	TS_DW_DATA_03
578 SYS_P1031	SYSAUX
579 PO	SYSAUX

Add partition.

```
183 |
184 | --Add partition
185 | ALTER TABLE calendar
186 |   ADD PARTITION PART_2 VALUES LESS THAN (TO_DATE('01/01/2015', 'DD/MM/YYYY'))
187 |   TABLESPACE ts_dw_data_02;
188 |
```

Script Output X
Task completed in 0.03 seconds

Table CALENDAR altered.

PARTITION (PART_2) is added.

The screenshot shows the SQL Worksheet interface with the following code and output:

```
188
189 SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
190
```

Script Output x | Query Result x

All Rows Fetched: 580 in 0.113 seconds

PARTITION_NAME	TABLESPACE_NAME
SYS_P1007	SYSAUX
SYS_P987	SYSAUX
SYS_P990	SYSAUX
SYS_P1027	SYSAUX
SYS_P1032	SYSAUX
SYS_P1028	SYSAUX
PART_1	TS_DW_DATA_03
SYS_P1031	SYSAUX
PART_2	TS_DW_DATA_02
P0	SYSAUX

Drop the table calendar.

The screenshot shows the SQL Worksheet interface with the following code and output:

```
195 drop table calendar;
```

Script Output x | Query Result x

Task completed in 0.044 seconds

Table CALENDAR dropped.

CREATE Example of Interval Composite partitioning

Create a table using Interval Composite partitioning

The screenshot shows the SQL Worksheet interface with the following code and output:

```
63
64 PARTITION BY RANGE (TIME_ID)
65   INTERVAL(NUMTODSINTERVAL(1, 'MONTH'))
66 (
67   PARTITION PART_1 VALUES LESS THAN (TO_DATE('01/01/2013', 'DD/MM/YYYY')),
68   PARTITION PART_2 VALUES LESS THAN (TO_DATE('01/07/2013', 'DD/MM/YYYY')),
69   PARTITION PART_3 VALUES LESS THAN (TO_DATE('01/01/2014', 'DD/MM/YYYY')),
70   PARTITION PART_4 VALUES LESS THAN (TO_DATE('01/07/2014', 'DD/MM/YYYY')),
71   PARTITION PART_5 VALUES LESS THAN (TO_DATE('01/01/2015', 'DD/MM/YYYY')),
72   PARTITION PART_6 VALUES LESS THAN (TO_DATE('01/07/2015', 'DD/MM/YYYY'))
73 )
74 TABLESPACE ts_dw_data_02;
```

Query Result x | Script Output x

Task completed in 0.058 seconds

Session altered.

Table CALENDAR created.

PARTITIONS (PART_1, PART_2, PART_3, PART_4, PART_5, PART_6) were created.

The screenshot shows the Oracle SQL Developer interface. A query window is open with the following SQL command:

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The results are displayed in a table titled "Script Output". The table has two columns: "PARTITION_NAME" and "TABLESPACE_NAME". The data shows 14 partitions, mostly named PART_1 through PART_6, which are located in the TS_DW_DATA_02 tablespace, except for some SYS_Pxxxx partitions which are in the SYSAUX tablespace.

PARTITION_NAME	TABLESPACE_NAME
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_6	TS_DW_DATA_02
580 PART_5	TS_DW_DATA_02
581 PART_4	TS_DW_DATA_02
582 SYS_P1051	SYSAUX
583 PART_3	TS_DW_DATA_02
584 PART_2	TS_DW_DATA_02
585 PART_1	TS_DW_DATA_02
586 P0	SYSAUX

Insert values into the table.

The screenshot shows the Oracle SQL Developer interface with a SQL Worksheet tab open. The worksheet contains the following SQL code:

```
143 | (
144 |   SELECT
145 |     TO_DATE( '12/31/2011', 'MM/DD/YYYY' ) sd,
146 |     rownum rn
147 |   FROM dual
148 |   CONNECT BY level <=1000
149 | );
150 |
```

The results of the execution are shown in the "Query Result" tab:

- Session altered.
- Table CALENDAR created.
- 1,000 rows inserted.

Merge partitions.

```
--Merge partitions

ALTER TABLE calendar
MERGE PARTITIONS PART_2, PART_3 INTO PARTITION PART_7;

Session altered.

Table CALENDAR created.

1,000 rows inserted.

Table CALENDAR altered.
```

SELECT * FROM calendar PARTITION (PART_7);										
TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	EB	
175 24-JUN-13	Monday	1	24	175	4	30-JUN-13	06	30	30-	
176 25-JUN-13	Tuesday	2	25	176	4	30-JUN-13	06	30	30-	
177 26-JUN-13	Wednesday	3	26	177	4	30-JUN-13	06	30	30-	
178 27-JUN-13	Thursday	4	27	178	4	30-JUN-13	06	30	30-	
179 28-JUN-13	Friday	5	28	179	4	30-JUN-13	06	30	30-	
180 29-JUN-13	Saturday	6	29	180	5	30-JUN-13	06	30	30-	
181 30-JUN-13	Sunday	7	30	181	5	30-JUN-13	06	30	30-	
182 01-JUL-13	Monday	1	01	182	1	07-JUL-13	07	31	31-	
183 02-JUL-13	Tuesday	2	02	183	1	07-JUL-13	07	31	31-	
184 03-JUL-13	Wednesday	3	03	184	1	07-JUL-13	07	31	31-	
185 04-JUL-13	Thursday	4	04	185	1	07-JUL-13	07	31	31-	
186 05-JUL-13	Friday	5	05	186	1	07-JUL-13	07	31	31-	
187 06-JUL-13	Saturday	6	06	187	1	07-JUL-13	07	31	31-	
188 07-JUL-13	Sunday	7	07	188	1	07-JUL-13	07	31	31-	
189 08-JUL-13	Monday	1	08	189	2	14-JUL-13	07	31	31-	
190 09-JUL-13	Tuesday	2	09	190	2	14-JUL-13	07	31	31-	
191 10-JUL-13	Wednesday	3	10	191	2	14-JUL-13	07	31	31-	

Move partition.

--Move partition	
ALTER TABLE calendar MOVE PARTITION PART_7 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS;	
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;	
PARTITION_NAME	TABLESPACE_NAME
566 SYS_P1008	SYSAUX
567 SYS_P180	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_6	TS_DW_DATA_02
580 PART_5	TS_DW_DATA_02
581 PART_4	TS_DW_DATA_02
582 SYS_P1051	SYSAUX
583 PART_1	TS_DW_DATA_02
584 PART_7	TS_DW_DATA_03
585 PO	SYSAUX

Split partition.

```
--Split partition
ALTER TABLE calendar SPLIT PARTITION PART_7
AT (TO_DATE('01/07/2013', 'DD/MM/YYYY'))
INTO
( PARTITION PART_8, PARTITION PART_9);

SELECT * FROM calendar PARTITION (PART_8);
SELECT * FROM calendar PARTITION (PART_9);

SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The screenshot shows the 'Query Result' tab of Oracle SQL Developer displaying the output of the 'ALL_TAB_PARTITIONS' view. The results are presented in a table with two columns: 'PARTITION_NAME' and 'TABLESPACE_NAME'. The data includes various system partitions like SYS_P966 through SYS_P1031, several temporary partitions (PART_1 through PART_9), and user-defined partitions such as TS_DW_DATA_02 and TS_DW_DATA_03. All partitions are located in the SYSAUX tablespace.

PARTITION_NAME	TABLESPACE_NAME
SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_6	TS_DW_DATA_02
580 PART_5	TS_DW_DATA_02
581 PART_4	TS_DW_DATA_02
582 SYS_P1051	SYSAUX
583 PART_9	TS_DW_DATA_03
584 PART_8	TS_DW_DATA_03
585 PART_1	TS_DW_DATA_02
586 PO	SYSAUX

Truncate partition.

```
--Truncate partition
ALTER TABLE calendar TRUNCATE PARTITION PART_9;
```

The screenshot shows the 'Query Result' tab of Oracle SQL Developer displaying the output of a TRUNCATE command. The message 'Task completed in 0.038 seconds' is shown, indicating the operation was successful.

Table CALENDAR truncated.

The screenshot shows the 'Query Result' tab of Oracle SQL Developer displaying the output of a SELECT query. The results are empty, as the PART_9 partition has been truncated. The table structure for the CALENDAR table is visible at the bottom of the screen.

Drop partition.

--Drop partition
ALTER TABLE calendar DROP PARTITION PART_9;

Script Output x Query Result x
Task completed in 0.029 seconds

Table CALENDAR altered.

SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;

Script Output x Query Result x
All Rows Fetched: 585 in 0.165 seconds

PARTITION_NAME	TABLESPACE_NAME
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_6	TS_DW_DATA_02
580 PART_5	TS_DW_DATA_02
581 PART_4	TS_DW_DATA_02
582 SYS_P1051	SYSAUX
583 PART_8	TS_DW_DATA_03
584 PART_1	TS_DW_DATA_02
585 P0	SYSAUX

Add partition.

Cannot perform the operation ADD PARTITION on an Interval partitioned object. Insert a row to create the new partition. If data is loaded into this table with a later date than the greatest defined partition, Oracle will automatically create a new partition for the new month.

```
--Add partition
ALTER TABLE calendar
  ADD PARTITION PART_8 VALUES LESS THAN (TO_DATE('01/01/2016', 'DD/MM/YYYY'))
  TABLESPACE ts_dw_data_02;
```

Script Output x | Task completed in 0.041 seconds

```
Error starting at line : 192 in command -
ALTER TABLE calendar
  ADD PARTITION PART_8 VALUES LESS THAN (TO_DATE('01/01/2016', 'DD/MM/YYYY'))
  TABLESPACE ts_dw_data_02
Error report -
ORA-14760: ADD PARTITION is not permitted on Interval partitioned objects
14760. 00000 -  "ADD PARTITION is not permitted on Interval partitioned objects"
*Cause:    ALTER TABLE ADD PARTITION was attempted on an Interval
           partitioned object.
*Action:   Do not perform the operation on an Interval partitioned object.
           Insert a row to create the new partition.
```

Drop the table calendar.

```
-- 
drop table calendar;
```

Script Output x | Task completed in 0.048 seconds

```
Table CALENDAR dropped.
```

CREATE Example of Hash partitioning

Create a table using Hash partitioning by CALENDAR_QUARTER_NUMBER.

The screenshot shows the VeraDB SQL Worksheet interface. The title bar displays 'Welcome Page', 'VeraDB', and 'Lab9_Task1_HASH partitioning.sql'. The main area is a 'Worksheet' tab showing the SQL code for creating a table:

```
END_OF_CAL_MONTH DATE,  
CALENDAR_MONTH_NAME VARCHAR2(32),  
DAYS_IN_CAL_QUARTER NUMBER,  
BEG_OF_CAL_QUARTER DATE,  
END_OF_CAL_QUARTER DATE,  
CALENDAR_QUARTER_NUMBER VARCHAR2(1),  
CALENDAR_YEAR VARCHAR2(4),  
DAYS_IN_CAL_YEAR NUMBER,  
BEG_OF_CAL_YEAR DATE,  
END_OF_CAL_YEAR DATE  
)  
  
PARTITION BY HASH (CALENDAR_QUARTER_NUMBER)  
  
(  
PARTITION PART_1,  
PARTITION PART_2  
)  
  
TABLESPACE ts_dw_data_02;
```

The code defines a table with columns for dates, month names, and quarter numbers, partitioned by hash based on the CALENDAR_QUARTER_NUMBER. The table is created in the 'ts_dw_data_02' tablespace.

The bottom pane shows the results of the query execution:

```
Grant succeeded.  
  
Session altered.  
  
Table CALENDAR created.
```

Insert values into the table.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, the active tab is "Lab9_Task1_HASH_partitioning.sql". The main area is a "Worksheet" tab showing the following PL/SQL code:

```
WHEN TO_CHAR( sd + rn, 'Q' ) = 2 THEN
    TO_DATE( '06/30/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
WHEN TO_CHAR( sd + rn, 'Q' ) = 3 THEN
    TO_DATE( '09/30/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
WHEN TO_CHAR( sd + rn, 'Q' ) = 4 THEN
    TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
END ) end_of_cal_quarter,
TO_CHAR( sd + rn, 'Q' ) calendar_quarter_number,
TO_CHAR( sd + rn, 'YYYY' ) calendar_year,
( TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
- TRUNC( sd + rn, 'YEAR' ) ) days_in_cal_year,
TRUNC( sd + rn, 'YEAR' ) beg_of_cal_year,
TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' ) end_of_cal_year
FROM
(
    SELECT
        TO_DATE( '12/31/2011', 'MM/DD/YYYY' ) sd,
        rownum rn
    FROM dual
    CONNECT BY level <=1000
);

```

Below the worksheet, the "Script Output" tab shows the results of the execution:

```
Session altered.

Table CALENDAR created.

1,000 rows inserted.
```

PARTITIONS (PART_1, PART_2) were created.

The screenshot shows the Oracle SQL Developer interface. A query has been run in the "Query Result" tab:

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The results are displayed in the "Script Output" tab:

```
Script Output | Query Result | All Rows Fetched: 582 in 0.132 seconds
```

PARTITION_NAME	TABLESPACE_NAME
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW_DATA_02
580 PART_1	TS_DW_DATA_02
581 SYS_P1051	SYSAUX
582 P0	SYSAUX

Move partition.

```
--Move partition

ALTER TABLE calendar MOVE PARTITION PART_1 TABLESPACE ts_dw_data_03;
```

Script Output x | Query Result x
Task completed in 0.102 seconds

Table CALENDAR altered.

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

Script Output x | Query Result x
SQL | All Rows Fetched: 582 in 0.147 seconds

PARTITION_NAME	TABLESPACE_...
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW_DATA_02
580 PART_1	TS_DW_DATA_03
581 SYS_P1051	SYSAUX
582 P0	SYSAUX

Truncate partition.

```
--Truncate partition

ALTER TABLE calendar TRUNCATE PARTITION PART_1;
```

Script Output x | Query Result x
Task completed in 0.038 seconds

Table CALENDAR truncated.

```
SELECT * FROM calendar PARTITION (PART_1);
```

Script Output x | Query Result x
SQL | All Rows Fetched: 0 in 0.011 seconds

TIME_ID	DAY_NAME	DAY_NUM...	DAY_NUM...	CALENDAR...	WEEK_EN...	CALENDAR...	DAYS_IN...	END_OF...	CALENDAR...	DAYS_IN...	BEG_OF...	END_OF...	CALENDAR...	...

Add partition.

```
--Add partition
ALTER TABLE calendar
    ADD PARTITION PART_3 TABLESPACE ts_dw_data_02;
ALTER TABLE calendar
    ADD PARTITION PART_4 TABLESPACE ts_dw_data_02;
```

Script Output x
Task completed in 0.037 seconds

Table CALENDAR altered.

Table CALENDAR altered.

PARTITIONS (PART_3, PART_4) were added.

The screenshot shows the Oracle SQL Developer interface. A query window is open with the following SQL command:

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The results are displayed in a table titled "Query Result". The columns are "PARTITION_NAME" and "TABLESPACE_NAME". The data shows various partitions, including PART_3 and PART_4 which were added, both located in the TS_DW_DATA_02 tablespace.

PARTITION_NAME	TABLESPACE_NAME
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW_DATA_02
580 PART_1	TS_DW_DATA_03
581 SYS_P1051	SYSAUX
582 PART_3	TS_DW_DATA_02
583 PART_4	TS_DW_DATA_02
584 PO	SYSAUX

Move partition Part_1 from tablespace ts_dw_data_03 to tablespace ts_dw_data_02.

The screenshot shows the Oracle SQL Developer interface. A script window contains the following SQL command:

```
--Move partition  
ALTER TABLE calendar MOVE PARTITION PART_1 TABLESPACE ts_dw_data_02;
```

The results are displayed in a table titled "Query Result". The columns are "PARTITION_NAME" and "TABLESPACE_NAME". The data shows that the PART_1 partition has been successfully moved to the TS_DW_DATA_02 tablespace.

PARTITION_NAME	TABLESPACE_NAME
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW_DATA_02
580 PART_1	TS_DW_DATA_02
581 SYS_P1051	SYSAUX
582 PART_3	TS_DW_DATA_02
583 PART_4	TS_DW_DATA_02
584 PO	SYSAUX

Coalesce partition.

All values from partitions before Coalesce partition.

SELECT * FROM calendar PARTITION (PART_1);									
TIME_ID	DAY_NAME	DAY_NUM...	DAY_NUM...	DAY_NUM...	CALENDAR...	WEEK_EN...	CALENDAR...	DAY_IN...	END_OF...

SELECT * FROM calendar PARTITION (PART_2);									
PF_CAL_MONTH	CALENDAR_MONTH_NAME	DAY_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER	CALENDAR_YEAR	DAY_IN_CAL_YEAR	BEG_OF_CAL_YEAR	
1-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
2-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
3-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
4-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
5-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
6-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
7-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
8-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
9-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
10-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
11-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
12-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
13-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
14-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
15-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
16-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
17-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
18-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
19-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
20-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
21-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
22-12	July	92 01-JUL-12	30-SEP-12	3	2012		365 01-JAN-12		
..									

SELECT * FROM calendar PARTITION (PART_3);									
TIME_ID	DAY_NAME	DAY_NUM...	DAY_NUM...	DAY_NUM...	CALENDAR...	WEEK_EN...	CALENDAR...	DAY_IN...	END_OF...

SELECT * FROM calendar PARTITION (PART_4);									
H_NUMBER	DAY_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DAY_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER	CALENDAR_YEAR	
1	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
2	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
3	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
4	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
5	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
6	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
7	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
8	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
9	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
10	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
11	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
12	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
13	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
14	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
15	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
16	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
17	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
18	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
19	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
20	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
21	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
22	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	
23	31	31-JAN-12	January		91 01-JAN-12	31-MAR-12	1	2012	

```
--COALESING PARTITIONS
ALTER TABLE calendar
COALESCE PARTITION;
```

Script Output | Task completed in 0.049 seconds

Table CALENDAR altered.

All values from partitions after Coalesce partition.

```
SELECT * FROM calendar PARTITION (PART_1);
```

Script Output | Query Result | All Rows Fetched: 0 in 0.01 seconds

TIME_ID	DAY_NAME	DAY_NUM...	DAY_NUM...	CALENDA...	WEEK_EN...	CALENDA...	DAYS_IN...	END_OF...	CALENDA...	DAYS_IN...	BEG_OF...	END_OF...	CALENDA...
265	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
266	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
267	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
268	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
269	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
270	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
271	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
272	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
273	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
274	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
275	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
276	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
277	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
278	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
279	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
280	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
281	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
282	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
283	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
284	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
285	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
286	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
287	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			


```
SELECT * FROM calendar PARTITION (PART_2);
```

Script Output | Query Result | Fetched 300 rows in 0.05 seconds

MONTH	CALENDAR_MONTH_NAME	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER	CALENDAR_YEAR	DAYS_IN_CAL_YEAR	BEG_OF_CAL_YEAR	EN
265	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
266	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
267	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
268	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
269	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
270	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
271	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
272	September	92 01-JUL-14	30-SEP-14	3	2014			364 01-JAN-14	31-I
273	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
274	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
275	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
276	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
277	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
278	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
279	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
280	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
281	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
282	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
283	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
284	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
285	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
286	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I
287	January	91 01-JAN-12	31-MAR-12	1	2012			365 01-JAN-12	31-I


```
SELECT * FROM calendar PARTITION (PART_3);
```

Script Output | Query Result | All Rows Fetched: 0 in 0.021 seconds

TIME_ID	DAY_NAME	DAY_NUM...	DAY_NUM...	CALENDA...	WEEK_EN...	CALENDA...	DAYS_IN...	END_OF...	CALENDA...	DAYS_IN...	BEG_OF...	END_OF...	CALENDA...
265	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
266	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
267	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
268	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
269	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
270	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
271	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
272	September	92 01-JUL-14	30-SEP-14	3	2014				364 01-JAN-14	31-I			
273	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
274	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
275	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
276	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
277	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
278	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
279	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
280	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
281	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
282	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
283	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
284	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
285	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
286	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			
287	January	91 01-JAN-12	31-MAR-12	1	2012				365 01-JAN-12	31-I			


```
SELECT * FROM calendar PARTITION (PART_4);
```

Script Output | Query Result | Executing:SELECT * FROM calendar PARTITION (PART_4) in 0 seconds

ORA-02149: Specified partition does not exist
02149. 00000 - "Specified partition does not exist %s"
*Cause: Partition not found for the object.
*Action: Retry with correct partition name.
Error at Line: 189 Column: 35

As we can notice, data from PART_4 coalesce with data from PART_2. As a result of the coalesce, PART_4 is dropped automatically.

The screenshot shows the SQL Developer interface with a query results window. The query executed is:

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The results show a list of partitions and their corresponding tablespaces. The columns are labeled "PARTITION_NAME" and "TABLESPACE_NAME".

PARTITION_NAME	TABLESPACE_NAME
562 SYS_P896	SYSAUX
563 SYS_P899	SYSAUX
564 SYS_P979	SYSAUX
565 SYS_P1003	SYSAUX
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW_DATA_02
580 PART_1	TS_DW_DATA_02
581 SYS_P1051	SYSAUX
582 PART_3	TS_DW_DATA_02
583 PO	SYSAUX

Drop the table calendar.

The screenshot shows the SQL Developer interface with a query results window. The query executed is:

```
--DROP  
drop table calendar;
```

The results show the table was altered and then dropped successfully.

Table CALENDAR altered.

Table CALENDAR dropped.

CREATE Example of List Composite partitioning

Create a table using List Composite partitioning.

The screenshot shows the VeraDB SQL Worksheet interface. The title bar displays 'Welcome Page', 'VeraDB', and 'Lab9_Task1_List Composite partitioning.sql'. The main area is a 'Worksheet' tab where the SQL code for creating the 'CALENDAR' table is pasted. The code defines a table with various columns and uses a 'PARTITION BY LIST' clause to partition it into four parts based on the month name. The table is created in the 'ts_dw_data_02' tablespace. Below the worksheet, a 'Script Output' tab shows the results of the execution: 'Session altered.' and 'Table CALENDAR created.'.

```
CREATE TABLE CALENDAR
(
    CALENDAR_WEEK_NUMBER           VARCHAR2(1),
    WEEK_ENDING_DATE               DATE,
    CALENDAR_MONTH_NUMBER          VARCHAR2(2),
    DAYS_IN_CAL_MONTH              VARCHAR2(2),
    END_OF_CAL_MONTH               DATE,
    CALENDAR_MONTH_NAME             VARCHAR2(32),
    DAYS_IN_CAL_QUARTER            NUMBER,
    BEG_OF_CAL_QUARTER              DATE,
    END_OF_CAL_QUARTER              DATE,
    CALENDAR_QUARTER_NUMBER         VARCHAR2(1),
    CALENDAR_YEAR                   VARCHAR2(4),
    DAYS_IN_CAL_YEAR                NUMBER,
    BEG_OF_CAL_YEAR                 DATE,
    END_OF_CAL_YEAR                 DATE
)
PARTITION BY LIST (CALENDAR_MONTH_NAME)
(
    PARTITION PART_1 VALUES ('December', 'January', 'February'),
    PARTITION PART_2 VALUES ('March', 'April', 'May'),
    PARTITION PART_3 VALUES ('June', 'July', 'August'),
    PARTITION PART_4 VALUES ('September', 'October', 'November')
)
TABLESPACE ts_dw_data_02;
```

Script Output

Session altered.

Table CALENDAR created.

Insert values into the table.

SQL Worksheet History

Worksheet Query Builder

```

    ( CASE
        WHEN TO_CHAR( sd + rn, 'Q' ) = 1 THEN
            TO_DATE( '03/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
        WHEN TO_CHAR( sd + rn, 'Q' ) = 2 THEN
            TO_DATE( '06/30/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
        WHEN TO_CHAR( sd + rn, 'Q' ) = 3 THEN
            TO_DATE( '09/30/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
        WHEN TO_CHAR( sd + rn, 'Q' ) = 4 THEN
            TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
    END ) end_of_cal_quarter,
    TO_CHAR( sd + rn, 'Q' ) calendar_quarter_number,
    TO_CHAR( sd + rn, 'YYYY' ) calendar_year,
    ( TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' )
    - TRUNC( sd + rn, 'YEAR' ) ) days_in_cal_year,
    TRUNC( sd + rn, 'YEAR' ) beg_of_cal_year,
    TO_DATE( '12/31/' || TO_CHAR( sd + rn, 'YYYY' ), 'MM/DD/YYYY' ) end_of_cal
)
FROM
(
    SELECT
        TO_DATE( '12/31/2011', 'MM/DD/YYYY' ) sd,
        rownum rn
    FROM dual
    CONNECT BY level <=1000
);

```

Script Output | Task completed in 0.072 seconds

Table CALENDAR created.

1,000 rows inserted.

SELECT * FROM calendar PARTITION (PART 1);

Script Output | Fetched 150 rows in 0.026 seconds

	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DAYS_IN_CAL_QUARTER	BEG
16 016	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
17 017	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
18 018	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
19 019	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
20 020	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
21 021	3	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
22 022	4	22-JAN-12	01	31	31-JAN-12	January		91 01-J-	
23 023	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
24 024	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
25 025	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
26 026	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
27 027	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
28 028	4	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
29 029	5	29-JAN-12	01	31	31-JAN-12	January		91 01-J-	
30 030	5	05-FEB-12	01	31	31-JAN-12	January		91 01-J-	
31 031	5	05-FEB-12	01	31	31-JAN-12	January		91 01-J-	
32 032	1	05-FEB-12	02	29	29-FEB-12	February		91 01-J-	
33 033	1	05-FEB-12	02	29	29-FEB-12	February		91 01-J-	
34 034	1	05-FEB-12	02	29	29-FEB-12	February		91 01-J-	
35 035	1	05-FEB-12	02	29	29-FEB-12	February		91 01-J-	
36 036	1	05-FEB-12	02	29	29-FEB-12	February		91 01-J-	
37 037	1	12-FEB-12	02	29	29-FEB-12	February		91 01-J-	
38 038	1	12-FEB-12	02	29	29-FEB-12	February		91 01-J-	
39 039	2	12-FEB-12	02	29	29-FEB-12	February		91 01-J-	

SELECT * FROM calendar PARTITION (PART_2);

	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DA
25	25	085	4	25-MAR-12	03	31	31-MAR-12	March	
26	26	086	4	01-APR-12	03	31	31-MAR-12	March	
27	27	087	4	01-APR-12	03	31	31-MAR-12	March	
28	28	088	4	01-APR-12	03	31	31-MAR-12	March	
29	29	089	5	01-APR-12	03	31	31-MAR-12	March	
30	30	090	5	01-APR-12	03	31	31-MAR-12	March	
31	31	091	5	01-APR-12	03	31	31-MAR-12	March	
32	01	092	1	01-APR-12	04	30	30-APR-12	April	
33	02	093	1	08-APR-12	04	30	30-APR-12	April	
34	03	094	1	08-APR-12	04	30	30-APR-12	April	
35	04	095	1	08-APR-12	04	30	30-APR-12	April	
36	05	096	1	08-APR-12	04	30	30-APR-12	April	
37	06	097	1	08-APR-12	04	30	30-APR-12	April	
38	07	098	1	08-APR-12	04	30	30-APR-12	April	
39	08	099	2	08-APR-12	04	30	30-APR-12	April	
40	09	100	2	15-APR-12	04	30	30-APR-12	April	
41	10	101	2	15-APR-12	04	30	30-APR-12	April	
42	11	102	2	15-APR-12	04	30	30-APR-12	April	
43	12	103	2	15-APR-12	04	30	30-APR-12	April	
44	13	104	2	15-APR-12	04	30	30-APR-12	April	
45	14	105	2	15-APR-12	04	30	30-APR-12	April	
46	15	106	3	15-APR-12	04	30	30-APR-12	April	
47	16	107	3	22-APR-12	04	30	30-APR-12	April	
48	17	108	3	22-APR-12	04	30	30-APR-12	April	
49	18	109	3	22-APR-12	04	30	30-APR-12	April	

SELECT * FROM calendar PARTITION (PART_3);

	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DA
22	22	174	4	24-JUN-12	06	30	30-JUN-12	June	
23	23	175	4	24-JUN-12	06	30	30-JUN-12	June	
24	24	176	4	24-JUN-12	06	30	30-JUN-12	June	
25	25	177	4	01-JUL-12	06	30	30-JUN-12	June	
26	26	178	4	01-JUL-12	06	30	30-JUN-12	June	
27	27	179	4	01-JUL-12	06	30	30-JUN-12	June	
28	28	180	4	01-JUL-12	06	30	30-JUN-12	June	
29	29	181	5	01-JUL-12	06	30	30-JUN-12	June	
30	30	182	5	01-JUL-12	06	30	30-JUN-12	June	
31	01	183	1	01-JUL-12	07	31	31-JUL-12	July	
32	02	184	1	08-JUL-12	07	31	31-JUL-12	July	
33	03	185	1	08-JUL-12	07	31	31-JUL-12	July	
34	04	186	1	08-JUL-12	07	31	31-JUL-12	July	
35	05	187	1	08-JUL-12	07	31	31-JUL-12	July	
36	06	188	1	08-JUL-12	07	31	31-JUL-12	July	
37	07	189	1	08-JUL-12	07	31	31-JUL-12	July	
38	08	190	2	08-JUL-12	07	31	31-JUL-12	July	
39	09	191	2	15-JUL-12	07	31	31-JUL-12	July	
40	10	192	2	15-JUL-12	07	31	31-JUL-12	July	
41	11	193	2	15-JUL-12	07	31	31-JUL-12	July	
42	12	194	2	15-JUL-12	07	31	31-JUL-12	July	
43	13	195	2	15-JUL-12	07	31	31-JUL-12	July	
44	14	196	2	15-JUL-12	07	31	31-JUL-12	July	
45	15	197	3	15-JUL-12	07	31	31-JUL-12	July	
46	16	198	3	22-JUL-12	07	31	31-JUL-12	July	

SELECT * FROM calendar PARTITION (PART_4);

	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DA
22	22	266	4	23-SEP-12	09	30	30-SEP-12	September	
23	23	267	4	23-SEP-12	09	30	30-SEP-12	September	
24	24	268	4	30-SEP-12	09	30	30-SEP-12	September	
25	25	269	4	30-SEP-12	09	30	30-SEP-12	September	
26	26	270	4	30-SEP-12	09	30	30-SEP-12	September	
27	27	271	4	30-SEP-12	09	30	30-SEP-12	September	
28	28	272	4	30-SEP-12	09	30	30-SEP-12	September	
29	29	273	5	30-SEP-12	09	30	30-SEP-12	September	
30	30	274	5	30-SEP-12	09	30	30-SEP-12	September	
31	01	275	1	07-OCT-12	10	31	31-OCT-12	October	
32	02	276	1	07-OCT-12	10	31	31-OCT-12	October	
33	03	277	1	07-OCT-12	10	31	31-OCT-12	October	
34	04	278	1	07-OCT-12	10	31	31-OCT-12	October	
35	05	279	1	07-OCT-12	10	31	31-OCT-12	October	
36	06	280	1	07-OCT-12	10	31	31-OCT-12	October	
37	07	281	1	07-OCT-12	10	31	31-OCT-12	October	
38	08	282	2	14-OCT-12	10	31	31-OCT-12	October	
39	09	283	2	14-OCT-12	10	31	31-OCT-12	October	
40	10	284	2	14-OCT-12	10	31	31-OCT-12	October	
41	11	285	2	14-OCT-12	10	31	31-OCT-12	October	
42	12	286	2	14-OCT-12	10	31	31-OCT-12	October	
43	13	287	2	14-OCT-12	10	31	31-OCT-12	October	
44	14	288	2	14-OCT-12	10	31	31-OCT-12	October	
45	15	289	3	21-OCT-12	10	31	31-OCT-12	October	
46	16	290	3	21-OCT-12	10	31	31-OCT-12	October	

Merge partitions.

```
--Merge partitions

ALTER TABLE calendar
MERGE PARTITIONS PART_2, PART_3 INTO PARTITION PART_5;

ALTER TABLE calendar
MERGE PARTITIONS PART_1, PART_4 INTO PARTITION PART_6;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a script editor window containing the two `ALTER TABLE` statements for merging partitions. Below it, the **Script Output** tab shows the results of the execution: "Table CALENDAR altered." twice. The **Query Result** tab shows the execution time as "Task completed in 0.086 seconds".

Move partition.

```
--Move partition

ALTER TABLE calendar MOVE PARTITION PART_5 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS;
ALTER TABLE calendar MOVE PARTITION PART_6 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a script editor window containing the two `ALTER TABLE` statements for moving partitions. Below it, the **Script Output** tab shows the results of the execution: "Table CALENDAR altered." twice. The **Query Result** tab shows the execution time as "Task completed in 0.107 seconds".

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a query editor window containing the `SELECT` statement to list all table partitions. Below it, the **Query Result** tab displays the results in a grid format:

PARTITION_NAME	TABLESPACE_NAME
579 SYS_P1051	SYSAUX
580 PART_5	TS_DW_DATA_03
581 PART_6	TS_DW_DATA_03
582 PO	SYSAUX

Split partition.

```
--Split partition
ALTER TABLE calendar SPLIT PARTITION PART_5
VALUES ('March', 'April', 'May')
INTO
( PARTITION PART_7, PARTITION PART_8);
```

Script Output X | Task completed in 0.05 seconds

Table CALENDAR altered.

```
SELECT * FROM calendar PARTITION (PART_7);
```

R_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER	CALENDAR
52	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
53	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
54	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
55	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
56	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
57	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
58	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
59	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
60	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
61	30	30-APR-12	April	91 01-APR-12	30-JUN-12	2	2012	
62	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
63	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
64	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
65	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
66	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
67	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
68	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
69	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	
70	31	31-MAY-12	May	91 01-APR-12	30-JUN-12	2	2012	

```
SELECT * FROM calendar PARTITION (PART_8);
```

M_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER	CALENDAR
58	31	31-JUL-12	July	92 01-JUL-12	30-SEP-12	3	2012	
59	31	31-JUL-12	July	92 01-JUL-12	30-SEP-12	3	2012	
60	31	31-JUL-12	July	92 01-JUL-12	30-SEP-12	3	2012	
61	31	31-JUL-12	July	92 01-JUL-12	30-SEP-12	3	2012	
62	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
63	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
64	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
65	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
66	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
67	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
68	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
69	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
70	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
71	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
72	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
73	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
74	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
75	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
76	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	
77	31	31-AUG-12	August	92 01-JUL-12	30-SEP-12	3	2012	

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

Script Output x Query Result x

SQL | All Rows Fetched: 583 in 0.169 seconds

PARTITION_NAME	TABLESPACE_NAME
562 SYS_P896	SYSAUX
563 SYS_P899	SYSAUX
564 SYS_P979	SYSAUX
565 SYS_P1003	SYSAUX
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_8	TS_DW_DATA_03
580 PART_7	TS_DW_DATA_03
581 SYS_P1051	SYSAUX
582 PART_6	TS_DW_DATA_03
583 P0	SYSAUX

Pay attention that when Part_5 was split into two parts, Part_5 was itself dropped.

Truncate partition.

```
--Truncate partition
ALTER TABLE calendar TRUNCATE PARTITION PART_6;
```

Script Output x Query Result x

Task completed in 0.031 seconds

Table CALENDAR altered.

Table CALENDAR truncated.

```
SELECT * FROM calendar PARTITION (PART_6);
```

Drop partition.

```
--Drop partition  
ALTER TABLE calendar DROP PARTITION PART_6;
```

Script Output x Query Result x
SQL | All Rows Fetched: 0 in 0.012 seconds

Table CALENDAR altered.

Table CALENDAR truncated.

Table CALENDAR altered.

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

Script Output x Query Result x
SQL | All Rows Fetched: 582 in 0.162 seconds

PARTITION_NAME	TABLESPACE_NAME
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_8	TS_DW_DATA_03
580 PART_7	TS_DW_DATA_03
581 SYS_P1051	SYSAUX
582 P0	SYSAUX

Add partition.

```
--Add partition  
ALTER TABLE calendar ADD PARTITION PART_1 VALUES ('December') TABLESPACE ts_dw_data_02;
```

Table CALENDAR altered.

```
SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
```

PARTITION_NAME	TABLESPACE_NAME
564 SYS_P979	SYSAUX
565 SYS_P1003	SYSAUX
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_8	TS_DW_DATA_03
580 PART_7	TS_DW_DATA_03
581 PART_1	TS_DW_DATA_02
582 SYS_P1051	SYSAUX
583 P0	SYSAUX

Drop the table calendar.

```
--  
drop table calendar;
```

Script Output x | Query Result x
| Task completed in 0.041 seconds

Table CALENDAR altered.
Table CALENDAR dropped.

CREATE Example of Reference partitioning

Create a calendar table with Range partitions.

```
Welcome Page x VeraDB x Lab9_Task1_Reference partitioning.sql x  
SQL Worksheet History  
Worksheet Query Builder  
56   END_OF_CAL_QUARTER           DATE,  
57   CALENDAR_QUARTER_NUMBER      VARCHAR2(1),  
58   CALENDAR_YEAR                VARCHAR2(4),  
59   DAYS_IN_CAL_YEAR            NUMBER,  
60   BEG_OF_CAL_YEAR              DATE,  
61   END_OF_CAL_YEAR             DATE,  
62   CONSTRAINT calendar_PK PRIMARY KEY (TIME_ID)  
63 )  
64  
65 PARTITION BY RANGE (TIME_ID)  
66  
67 (  
68 PARTITION PART_1 VALUES LESS THAN (TO_DATE('01/01/2013', 'DD/MM/Y  
69 PARTITION PART_2 VALUES LESS THAN (TO_DATE('01/01/2015', 'DD/MM/Y  
70 )  
71  
72 TABLESPACE ts_dw_data_02;
```

Script Output x
| Task completed in 0.054 seconds

Table CALENDAR created.

Insert values into the table.

The screenshot shows a SQL Worksheet window with the following details:

- Tab Bar:** Welcome Page, VeraDB, Lab9_Task1_Reference partitioning.sql
- Toolbar:** SQL Worksheet, History, various icons for file operations, and a status bar showing "0.063 seconds".
- Worksheet Tab:** Worksheet (selected), Query Builder
- Code Area:** A block of PL/SQL code for creating a CALENDAR table. The code uses dynamic SQL to generate rows for each day of the year 2011, partitioned by year. Lines 134-146 show the main logic, and line 147 shows the final command.
- Script Output Tab:** Shows the output of the script execution:
 - Table CALENDAR created.
 - 1,000 rows inserted.

Create a table using Reference partitioning.

The screenshot shows a SQL Worksheet window with the following details:

- Tab Bar:** Welcome Page, VeraDB, Lab9_Task1_Reference partitioning.sql
- Toolbar:** SQL Worksheet, History, various icons for file operations, and a status bar showing "0.105 seconds".
- Worksheet Tab:** Worksheet (selected), Query Builder
- Code Area:** A block of PL/SQL code for creating an ORDER_CALENDAR table. The table has columns for order_id, TIME_ID, line_item_id, product_id, unit_price, and quantity. It includes a foreign key constraint TIME_ID_fk referencing the TIME_ID column of the calendar table, and is partitioned by reference using the same partitioning scheme as the calendar table. Line 162 shows the TABLESPACE assignment.
- Script Output Tab:** Shows the output of the script execution:
 - Table CALENDAR created.
 - 1,000 rows inserted.
 - Table ORDER_CALENDAR created.

Move partition.

```
164 --Move partition
165
166 ALTER TABLE order_calendar MOVE PARTITION PART_2 TABLESPACE ts_dw_data_03 NOLOGGING COMPRESS;
167
```

Script Output | Query Result | Task completed in 0.049 seconds

User U_DW_DATA_03 altered.

Session altered.

Grant succeeded.

Session altered.

Table CALENDAR created.

1,000 rows inserted.

Table ORDER_CALENDAR created.

Table ORDER_CALENDAR altered.

```
167
168 SELECT partition_name, tablespace_name FROM ALL_TAB_PARTITIONS;
169
```

Script Output | Query Result | All Rows Fetched: 584 in 0.127 seconds

PARTITION_NAME	TABLESPACE_NAME
565 SYS_P1003	SYSAUX
566 SYS_P1008	SYSAUX
567 SYS_P980	SYSAUX
568 SYS_P963	SYSAUX
569 SYS_P966	SYSAUX
570 SYS_P1004	SYSAUX
571 SYS_P1007	SYSAUX
572 SYS_P987	SYSAUX
573 SYS_P990	SYSAUX
574 SYS_P1027	SYSAUX
575 SYS_P1032	SYSAUX
576 SYS_P1052	SYSAUX
577 SYS_P1028	SYSAUX
578 SYS_P1031	SYSAUX
579 PART_2	TS_DW
580 PART_1	TS_DW
581 SYS_P1051	SYSAUX
582 PART_2	TS_DW
583 PART_1	TS_DW
584 P0	SYSAUX

Truncate partition.

The screenshot shows a database editor window with several lines of SQL code. Line 170 contains a comment '--Truncate partition'. Line 171 contains the command 'ALTER TABLE order_calendar TRUNCATE PARTITION PART_2;'. Line 172 is a blank line. Below the editor is a 'Script Output' window with a toolbar. The output pane shows the message 'Table ORDER CALENDAR truncated.' followed by a timestamp 'Task completed in 0.039 seconds'.

```
--Truncate partition
ALTER TABLE order_calendar TRUNCATE PARTITION PART_2;

```

Table ORDER CALENDAR truncated.
Task completed in 0.039 seconds

Drop the table order_calendar and the table calendar.

The screenshot shows a database editor window with several lines of SQL code. Lines 175 through 179 contain the following commands: '--DROP', 'drop table order_calendar;', 'drop table calendar;', 'drop TABLESPACE ts_dw_data_02 INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;', and 'drop TABLESPACE ts_dw_data_03 INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;'. Below the editor is a 'Script Output' window with a toolbar. The output pane shows five messages: 'Table ORDER CALENDAR dropped.', 'Table CALENDAR dropped.', 'TABLESPACE TS DW DATA 02 dropped.', and 'TABLESPACE TS DW DATA 03 dropped.' each followed by a timestamp.

```
--DROP
drop table order_calendar;
drop table calendar;
drop TABLESPACE ts_dw_data_02 INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;
drop TABLESPACE ts_dw_data_03 INCLUDING CONTENTS AND DATAFILES CASCADE CONSTRAINTS;
```

Table ORDER CALENDAR dropped.
Table CALENDAR dropped.
TABLESPACE TS DW DATA 02 dropped.
TABLESPACE TS DW DATA 03 dropped.

3. Business Task - Partitioning Facts

3.1. Task 02: Partitioning Facts

Partitioning Fact table.

Partitioning is done to enhance performance and facilitate easy management of data. Partitioning also helps in balancing the various requirements of the system. It optimizes the hardware performance and simplifies the management of data warehouse by partitioning fact table into multiple separate partitions.

In terms of the usability of assessing data from FACT_TABLE, it is suggested to use data partitioning by quarters in the context of network agencies.

Change Creation Script for Fact Table.

Welcome Page VeraDB t_FCT_business.sql

SQL Worksheet History

Worksheet Query Builder

```

7   TIME_ID           DATE,
8   customer_ID       NUMBER(10),
9   employee_ID       NUMBER(10),
10  agency_ID         NUMBER(10),
11  gen_period_ID    NUMBER(10),
12  product_ID        NUMBER(10),
13  promotion_ID     NUMBER(10),
14  gross_profit_dollar_amount DECIMAL (30,2),
15  net_profit_dollar_amount DECIMAL (30,2),
16  gross_revenue_dollar_amount DECIMAL (30,2),
17  net_revenue_dollar_amount DECIMAL (30,2),
18  gross_cost_dollar_amount DECIMAL (30,2),
19  net_cost_dollar_amount DECIMAL (30,2),
20  gross_salary_employee_dollar_amount DECIMAL (30,2),
21  net_salary_employee_dollar_amount DECIMAL (30,2),
22  customer_quantity  DECIMAL (30,2),
23  employee_quantity  DECIMAL (30,2),
24  promotion_quantity DECIMAL (30,2),
25  revenue_cost_percent DECIMAL (10,2),
26  CONSTRAINT "PK_T.FCT_business" PRIMARY KEY(Business_Fact_ID)
27 )
28 PARTITION BY RANGE (TIME_ID)
29   subpartition by hash(agency_ID ) subpartitions 4
30 (
31   PARTITION QUARTER_1 VALUES LESS THAN(TO_DATE('01/04/2022', 'DD/MM/YYYY'))
32   ( subpartition QUARTER_1_sub_1,
33     subpartition QUARTER_1_sub_2,
34     subpartition QUARTER_1_sub_3,
35     subpartition QUARTER_1_sub_4
36   ),

```

Script Output x | Task completed in 0.099 seconds

Table FCT_BUSINESS created.

Partitioning scheme:

