CREATE USER anj\_prac3 IDENTIFIED BY 123456;

GRANT CONNECT TO anj\_prac3;

GRANT CONNECT, RESOURCE, DBA TO anj\_prac3;

GRANT UNLIMITED TABLESPACE TO anj\_prac3;

Q.1

CREATE TABLE sales(

prod\_id NUMBER,

cust\_id NUMBER,

promo\_id NUMBER,

quantity\_sold NUMBER,

amount\_sold NUMBER,

time\_id DATE

)

PARTITION BY RANGE (time\_id)(

PARTITION sales\_q1 VALUES LESS THAN (TO\_DATE('01-APR-2017','dd-MON-yyyy'))

TABLESPACE tsa,

PARTITION sales\_q2 VALUES LESS THAN (TO\_DATE('01-JUL-2017','dd-MON-yyyy'))

TABLESPACE tsb,

PARTITION sales\_q3 VALUES LESS THAN (TO\_DATE('01-OCT-2017','dd-MON-yyyy'))

TABLESPACE tsc,

PARTITION sales\_q4 VALUES LESS THAN (TO\_DATE('01-JAN-2018','dd-MON-yyyy'))

TABLESPACE tsd

);

Table created.

INSERT INTO sales VALUES(1,1,1,2,2,'22-JAN-2016');

1 row created.

SQL> select \* from sales partition(sales\_q1);

PROD\_ID CUST\_ID PROMO\_ID QUANTITY\_SOLD AMOUNT\_SOLD TIME\_ID

---------- ---------- ---------- ------------- ----------- ---------

1 1 1 2 2 22-JAN-16

SQL> select \* from sales partition(sales\_q2);

no rows selected

SQL> select \* from sales partition(sales\_q3);

no rows selected

SQL> select \* from sales partition(sales\_q4);

no rows selected

SQL> UPDATE sales

2 SET time\_id='02-JUN-2017'

3 WHERE prod\_id=1;

UPDATE sales

\*

ERROR at line 1:

ORA-14402: updating partition key column would cause a partition change

Q.2

CREATE TABLE sales\_rm(

prod\_id NUMBER,

cust\_id NUMBER,

promo\_id NUMBER,

quantity\_sold NUMBER,

amount\_sold NUMBER,

time\_id DATE

)

PARTITION BY RANGE (time\_id)(

PARTITION sales\_q1 VALUES LESS THAN (TO\_DATE('01-APR-2017','dd-MON-yyyy'))

TABLESPACE tsa,

PARTITION sales\_q2 VALUES LESS THAN (TO\_DATE('01-JUL-2017','dd-MON-yyyy'))

TABLESPACE tsb,

PARTITION sales\_q3 VALUES LESS THAN (TO\_DATE('01-OCT-2017','dd-MON-yyyy'))

TABLESPACE tsc,

PARTITION sales\_q4 VALUES LESS THAN (TO\_DATE('01-JAN-2018','dd-MON-yyyy'))

TABLESPACE tsd

)

ENABLE ROW MOVEMENT;

Table created.

SQL> INSERT INTO sales\_rm VALUES(1,1,1,2,2,'22-JAN-2016');

1 row created.

SQL> UPDATE sales\_rm

2 SET time\_id='02-JUN-2017'

3 WHERE prod\_id=1;

1 row updated.

Q.3

CREATE TABLE dept(

deptno number,

deptname varchar(50),

quarterly\_sales number,

state varchar2(2))

PARTITION BY LIST (state)(

PARTITION northwest VALUES ('OR', 'WA'),

PARTITION southwest VALUES ('AZ', 'UT', 'NM'),

PARTITION northeast VALUES ('NY', 'VM', 'NJ'),

PARTITION southeast VALUES ('FL', 'GA'),

PARTITION northcentral VALUES ('SD', 'WI'),

PARTITION southcentral VALUES ('OK', 'TX')

);

Table created.

INSERT INTO dept VALUES(10, 'accounting', 100, 'WA');

1 row created.

SQL> insert into dept values(&deptno,'&deptname',&quarterly\_sales,'&state');

Enter value for deptno: 20

Enter value for deptname: R&D

Enter value for quarterly\_sales: 150

Enter value for state: OR

old 1: insert into dept values(&deptno,'&deptname',&quarterly\_sales,'&state')

new 1: insert into dept values(20,'R&D',150,'OR')

1 row created.

SQL> /

Enter value for deptno: 30

Enter value for deptname: sales

Enter value for quarterly\_sales: 100

Enter value for state: FL

old 1: insert into dept values(&deptno,'&deptname',&quarterly\_sales,'&state')

new 1: insert into dept values(30,'sales',100,'FL')

1 row created.

SQL> /

Enter value for deptno: 40

Enter value for deptname: HR

Enter value for quarterly\_sales: 10

Enter value for state: TX

old 1: insert into dept values(&deptno,'&deptname',&quarterly\_sales,'&state')

new 1: insert into dept values(40,'HR',10,'TX')

1 row created.

SQL> /

Enter value for deptno: 50

Enter value for deptname: systems engineering

Enter value for quarterly\_sales: 10

Enter value for state: CA

old 1: insert into dept values(&deptno,'&deptname',&quarterly\_sales,'&state')

new 1: insert into dept values(50,'systems engineering',10,'CA')

insert into dept values(50,'systems engineering',10,'CA')

\*

ERROR at line 1:

ORA-14400: inserted partition key does not map to any partition

SQL> SELECT \* FROM dept PARTITION(northwest);

DEPTNO DEPTNAME QUARTERLY\_SALES ST

---------- -------------------------------------------------- --------------- --

10 accounting 100 WA

20 R&D 150 OR

SQL> SELECT \* FROM dept PARTITION(southwest);

no rows selected

SQL> SELECT \* FROM dept PARTITION(northeast);

no rows selected

SQL> SELECT \* FROM dept PARTITION(southeast);

DEPTNO DEPTNAME QUARTERLY\_SALES ST

---------- -------------------------------------------------- --------------- --

30 sales 100 FL

SQL> SELECT \* FROM dept PARTITION(northcentral);

no rows selected

SQL> SELECT \* FROM dept PARTITION(southcentral);

DEPTNO DEPTNAME QUARTERLY\_SALES ST

---------- -------------------------------------------------- --------------- --

40 HR 10 TX

Q.4

SQL> CREATE TABLE dates(

2 year NUMBER,

3 month NUMBER,

4 day NUMBER,

5 amount\_sold NUMBER)

6 PARTITION BY RANGE (year,month) (

7 PARTITION before\_2001 VALUES LESS THAN (2001,1),

8 PARTITION apr\_2001 VALUES LESS THAN (2001,4),

9 PARTITION july\_2001 VALUES LESS THAN (2001,7),

10 PARTITION oct\_2001 VALUES LESS THAN (2001,10),

11 PARTITION jan\_2002 VALUES LESS THAN (2002,1),

12 PARTITION future VALUES LESS THAN (MAXVALUE,0)

13 );

Table created.

INSERT INTO dates VALUES(2001,3,17, 2000);

1 row created.

INSERT INTO dates VALUES(2001,11,1, 5000);

1 row created.

INSERT INTO dates VALUES(2002,1,1, 4000);

1 row created.

SQL> select \* from dates partition(before\_2001);

no rows selected

SQL> select \* from dates partition(apr\_2001);

YEAR MONTH DAY AMOUNT\_SOLD

---------- ---------- ---------- -----------

2001 3 17 2000

SQL> select \* from dates partition(july\_2001);

no rows selected

SQL> select \* from dates partition(oct\_2001);

no rows selected

SQL> select \* from dates partition(jan\_2002);

YEAR MONTH DAY AMOUNT\_SOLD

---------- ---------- ---------- -----------

2001 11 1 5000

SQL> select \* from dates partition(future);

YEAR MONTH DAY AMOUNT\_SOLD

---------- ---------- ---------- -----------

2002 1 1 4000

Q.5

SQL> CREATE TABLE supplier\_parts(

2 supplier\_id NUMBER,

3 partnum NUMBER,

4 price NUMBER)

5 PARTITION BY RANGE (supplier\_id, partnum)(

6 PARTITION p1 VALUES LESS THAN (10,100),

7 PARTITION p2 VALUES LESS THAN (10,200),

8 PARTITION p3 VALUES LESS THAN (MAXVALUE,MAXVALUE));

Table created.

SQL> INSERT INTO supplier\_parts VALUES(&supplier\_id,&partnum,&price);

Enter value for supplier\_id: 5

Enter value for partnum: 5

Enter value for price: 1000

old 1: INSERT INTO supplier\_parts VALUES(&supplier\_id,&partnum,&price)

new 1: INSERT INTO supplier\_parts VALUES(5,5,1000)

1 row created.

SQL> /

Enter value for supplier\_id: 5

Enter value for partnum: 150

Enter value for price: 1000

old 1: INSERT INTO supplier\_parts VALUES(&supplier\_id,&partnum,&price)

new 1: INSERT INTO supplier\_parts VALUES(5,150,1000)

1 row created.

SQL> /

Enter value for supplier\_id: 10

Enter value for partnum: 100

Enter value for price: 1000

old 1: INSERT INTO supplier\_parts VALUES(&supplier\_id,&partnum,&price)

new 1: INSERT INTO supplier\_parts VALUES(10,100,1000)

1 row created.

SQL> SELECT \* FROM supplier\_parts PARTITION(p1);

SUPPLIER\_ID PARTNUM PRICE

----------- ---------- ----------

5 5 1000

5 150 1000

SQL> SELECT \* FROM supplier\_parts PARTITION(p2);

SUPPLIER\_ID PARTNUM PRICE

----------- ---------- ----------

10 100 1000

Q.6

SQL> CREATE TABLE int\_sales(

2 prod\_id NUMBER,

3 cust\_id NUMBER,

4 time\_id DATE,

5 promo\_id NUMBER,

6 quantity\_sold NUMBER,

7 amount\_sold NUMBER)

8 PARTITION BY RANGE (time\_id)

9 INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))(

10 PARTITION p0 VALUES LESS THAN (TO\_DATE('1-4-2009', 'DD-MM-YYYY')),

11 PARTITION p1 VALUES LESS THAN (TO\_DATE('1-7-2009', 'DD-MM-YYYY')),

12 PARTITION p2 VALUES LESS THAN (TO\_DATE('1-10-2009', 'DD-MM-YYYY')),

13 PARTITION p3 VALUES LESS THAN (TO\_DATE('1-1-2010', 'DD-MM-YYYY')) );

Table created.

SQL> INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold);

Enter value for prod\_id: 10

Enter value for cust\_id: 0

Enter value for time\_id: 20-feb-2010

Enter value for promo\_id: 40

Enter value for quantity\_sold: 6

Enter value for amount\_sold: 6

old 1: INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold)

new 1: INSERT INTO int\_sales VALUES(10,0,'20-feb-2010',40,6,6)

1 row created.

SQL> INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold);

Enter value for prod\_id: 4

Enter value for cust\_id: 4

Enter value for time\_id: 1-may-2009

Enter value for promo\_id: 5

Enter value for quantity\_sold: 5

Enter value for amount\_sold: 5

old 1: INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold)

new 1: INSERT INTO int\_sales VALUES(4,4,'1-may-2009',5,5,5)

1 row created.

SQL> INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold);

Enter value for prod\_id: 8

Enter value for cust\_id: 8

Enter value for time\_id: 28-mar-2010

Enter value for promo\_id: 7

Enter value for quantity\_sold: 7

Enter value for amount\_sold: 7

old 1: INSERT INTO int\_sales VALUES(&prod\_id,&cust\_id,'&time\_id',&promo\_id,&quantity\_sold,&amount\_sold)

new 1: INSERT INTO int\_sales VALUES(8,8,'28-mar-2010',7,7,7)

1 row created.

SQL> desc user\_tab\_partitions;

Name Null? Type

----------------------------------------- -------- ----------------------------

TABLE\_NAME VARCHAR2(30)

COMPOSITE VARCHAR2(3)

PARTITION\_NAME VARCHAR2(30)

SUBPARTITION\_COUNT NUMBER

HIGH\_VALUE LONG

HIGH\_VALUE\_LENGTH NUMBER

PARTITION\_POSITION NUMBER

TABLESPACE\_NAME VARCHAR2(30)

PCT\_FREE NUMBER

PCT\_USED NUMBER

INI\_TRANS NUMBER

MAX\_TRANS NUMBER

INITIAL\_EXTENT NUMBER

NEXT\_EXTENT NUMBER

MIN\_EXTENT NUMBER

MAX\_EXTENT NUMBER

MAX\_SIZE NUMBER

PCT\_INCREASE NUMBER

FREELISTS NUMBER

FREELIST\_GROUPS NUMBER

LOGGING VARCHAR2(7)

COMPRESSION VARCHAR2(8)

COMPRESS\_FOR VARCHAR2(12)

NUM\_ROWS NUMBER

BLOCKS NUMBER

EMPTY\_BLOCKS NUMBER

AVG\_SPACE NUMBER

CHAIN\_CNT NUMBER

AVG\_ROW\_LEN NUMBER

SAMPLE\_SIZE NUMBER

LAST\_ANALYZED DATE

BUFFER\_POOL VARCHAR2(7)

FLASH\_CACHE VARCHAR2(7)

CELL\_FLASH\_CACHE VARCHAR2(7)

GLOBAL\_STATS VARCHAR2(3)

USER\_STATS VARCHAR2(3)

IS\_NESTED VARCHAR2(3)

PARENT\_TABLE\_PARTITION VARCHAR2(30)

INTERVAL VARCHAR2(3)

SEGMENT\_CREATED VARCHAR2(3)

SQL> select table\_name,partition\_name

2 from user\_tab\_partitions

3 where table\_name='int\_sales';

no rows selected

SQL> select table\_name,partition\_name

2 from USER\_TAB\_PARTITIONS

3 where table\_name='INT\_SALES';

TABLE\_NAME PARTITION\_NAME

------------------------------ ------------------------------

INT\_SALES P0

INT\_SALES P1

INT\_SALES P2

INT\_SALES P3

INT\_SALES SYS\_P49

INT\_SALES SYS\_P50

6 rows selected.

SQL> SELECT \* FROM INT\_SALES PARTITION(SYS\_P49);

PROD\_ID CUST\_ID TIME\_ID PROMO\_ID QUANTITY\_SOLD AMOUNT\_SOLD

---------- ---------- --------- ---------- ------------- -----------

10 0 20-FEB-10 40 6 6

SQL> SELECT \* FROM INT\_SALES PARTITION(SYS\_P50);

PROD\_ID CUST\_ID TIME\_ID PROMO\_ID QUANTITY\_SOLD AMOUNT\_SOLD

---------- ---------- --------- ---------- ------------- -----------

8 8 28-MAR-10 7 7 7

SQL> SELECT \* FROM INT\_SALES PARTITION(P1);

PROD\_ID CUST\_ID TIME\_ID PROMO\_ID QUANTITY\_SOLD AMOUNT\_SOLD

---------- ---------- --------- ---------- ------------- -----------

4 4 01-MAY-09 5 5 5

Q.7

SQL> CREATE TABLE orders(

2 order\_id NUMBER,

3 order\_date DATE,

4 customer\_id NUMBER,

5 shipper\_id NUMBER,

6 CONSTRAINT orders\_pk PRIMARY KEY(order\_id)

7 )

8 PARTITION BY RANGE(order\_date)(

9 PARTITION Q1 VALUES LESS THAN (TO\_DATE('01-APR-2008','DD-MON-YYYY')),

10 PARTITION Q2 VALUES LESS THAN (TO\_DATE('01-JUL-2008','DD-MON-YYYY')),

11 PARTITION Q3 VALUES LESS THAN (TO\_DATE('01-OCT-2008','DD-MON-YYYY')),

12 PARTITION Q4 VALUES LESS THAN (TO\_DATE('01-JAN-2009','DD-MON-YYYY'))

13 );

Table created.

SQL> CREATE TABLE order\_items(

2 order\_id NUMBER NOT NULL,

3 product\_id NUMBER NOT NULL,

4 unit\_price NUMBER,

5 quantity NUMBER,

6 CONSTRAINT order\_items\_fk FOREIGN KEY(order\_id) REFERENCES orders(order\_id)

7 )

8 PARTITION BY REFERENCE(order\_items\_fk);

Table created.

Q8

SQL> CREATE TABLE EMPLOYEE

2 (

3 emp\_id NUMBER NOT NULL,

4 emp\_name VARCHAR(50),

5 fixed\_salary NUMBER NOT NULL,

6 variable\_salary NUMBER NOT NULL,

7 total\_salary AS (fixed\_salary + variable\_salary)

8 )

9 PARTITION BY RANGE(total\_salary)

10 (

11 PARTITION p1 VALUES LESS THAN (25000),

12 PARTITION p2 VALUES LESS THAN (50000),

13 PARTITION p3 VALUES LESS THAN (75000),

14 PARTITION p4 VALUES LESS THAN (MAXVALUE)

15 );

Table created.

Q9

SQL> CREATE TABLE customer(  
  2  cust\_id number,  
  3  cust\_name varchar(20),  
  4  cust\_state varchar(5),  
  5  time\_id date)  
  6  PARTITION BY RANGE (time\_id)  
  7  SUBPARTITION BY LIST (cust\_state)  
  8  (PARTITION old VALUES LESS THAN (TO\_DATE('1-JAN-2005','DD-MON-YYYY'))  
  9           (SUBPARTITION old\_west VALUES ('MH', 'GJ'),  
 10            SUBPARTITION old\_south VALUES ('TN', 'AP'),  
 11            SUBPARTITION old\_north VALUES ('UP', 'HP'),  
 12            SUBPARTITION old\_others VALUES (DEFAULT)  
 13           ),  
 14  PARTITION acquired VALUES LESS THAN (TO\_DATE('1-JAN-2010','DD-MON-YYYY'))  
 15           (SUBPARTITION acquired\_west VALUES ('MH', 'GJ'),  
 16            SUBPARTITION acquired\_south VALUES ('TN', 'AP'),  
 17            SUBPARTITION acquired\_north VALUES ('UP', 'HP'),  
 18            SUBPARTITION acquired\_others VALUES (DEFAULT)  
 19           ),  
 20  PARTITION recent VALUES LESS THAN (TO\_DATE('1-JAN-2015','DD-MON-YYYY'))  
 21           (SUBPARTITION recent\_west VALUES ('MH', 'GJ'),  
 22            SUBPARTITION recent\_south VALUES ('TN', 'AP'),  
 23            SUBPARTITION recent\_north VALUES ('UP', 'HP'),  
 24            SUBPARTITION recent\_others VALUES (DEFAULT)  
 25           ),  
 26  PARTITION unknown VALUES LESS THAN (MAXVALUE)  
 27           (SUBPARTITION unknown\_west VALUES ('MH', 'GJ'),  
 28            SUBPARTITION unknown\_south VALUES ('TN', 'AP'),  
 29            SUBPARTITION unknown\_north VALUES ('UP', 'HP'),  
 30            SUBPARTITION unknown\_others VALUES (DEFAULT)  
 31           )  
 32  );  
  
Table created.