PARTITIONS

**Partitioning** enables you to decompose very large tables and indexes into smaller and more manageable pieces called **partitions**. Each partition is an independent object with its own name and optionally its own storage characteristics.

Advantages:

1. Manageability
2. Scalability
3. Availability

**Partitions are 3 types**

1. **Range partition**
2. **List partition**
3. **Hash partition**

**Range Partition**:

Created based on range of values from a column.

CREATE TABLE time\_range\_sales

(

prod\_id NUMBER(6)

, cust\_id NUMBER

, **time\_id** DATE

, channel\_id CHAR(1)

, promo\_id NUMBER(6)

, quantity\_sold NUMBER(3)

, amount\_sold NUMBER(10,2)

)

PARTITION BY RANGE (**time\_id**)

(PARTITION SALES\_2014 VALUES LESS THAN

(TO\_DATE('01-JAN-2015','DD-MON-YYYY')),

PARTITION SALES\_2015 VALUES LESS THAN

(TO\_DATE('01-JAN-2016','DD-MON-YYYY')),

PARTITION SALES\_2016 VALUES LESS THAN

(TO\_DATE('01-JAN-2017','DD-MON-YYYY')),

PARTITION SALES\_2017 VALUES LESS THAN

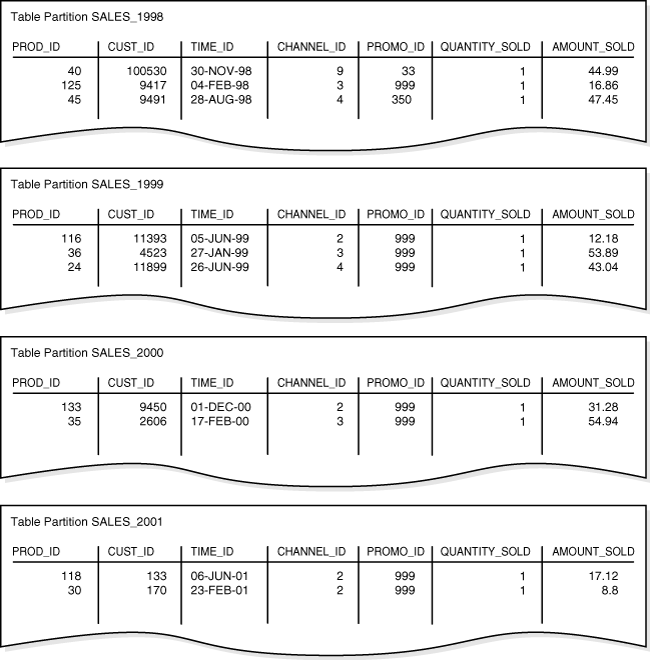
(TO\_DATE('01-JAN-2018','DD-MON-YYYY'),

PARTITION SALES\_2018 VALUES LESS THAN

(TO\_DATE('01-JAN-2019','DD-MON-YYYY')

);

***Figure 4-1 Range Partitions***



**List Partitioning:**

It is created based on list of values from a column.

CREATE TABLE list\_sales

(

prod\_id NUMBER(6)

, cust\_id NUMBER

, time\_id DATE

, **channel\_id** CHAR(1)

, promo\_id NUMBER(6)

, quantity\_sold NUMBER(3)

, amount\_sold NUMBER(10,2)

)

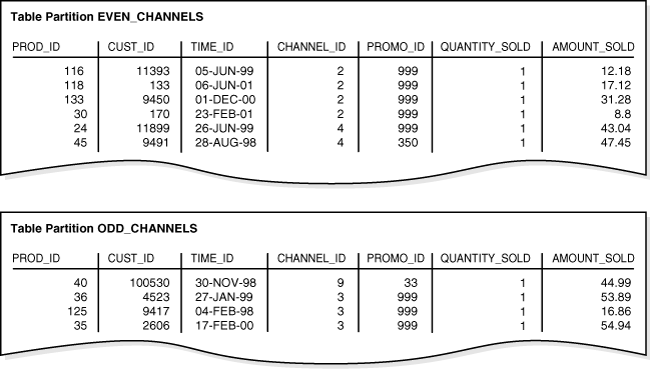
PARTITION BY LIST (**channel\_id**)

(

PARTITION even\_channels VALUES (2,4),

PARTITION odd\_channels VALUES (3,9)

);



**HASH Partitioning:**

It is created based on equal size, means each partition contains same number of records.

CREATE TABLE hash\_sales

(

**prod\_id** NUMBER(6)

, cust\_id NUMBER

, time\_id DATE

, channel\_id CHAR(1)

, promo\_id NUMBER(6)

, quantity\_sold NUMBER(3)

, amount\_sold NUMBER(10,2)

)

PARTITION BY HASH (**prod\_id**)

PARTITIONS 2;

***Figure 4-3 Hash Partitions***

