Table Paritioning

Example:

Create Table:

create table bookings(flightno varchar(200),flightname varchar(200),booking\_date timestamp);

create table jan\_bookings(check(booking\_date >= date '2020-01-01' and booking\_date <= '2020-01-31')) inherits(bookings);

create table feb\_bookings(check(booking\_date >= date '2020-02-01' and booking\_date <= '2020-02-29')) inherits(bookings);

nano=# \d+ bookings

Table "public.bookings"

Column | Type | Collation | Nullable | Default | Storage | Stats target | Description

--------------+-----------------------------+-----------+----------+---------+----------+--------------+-------------

flightno | character varying(200) | | | | extended | |

flightname | character varying(200) | | | | extended | |

booking\_date | timestamp without time zone | | | | plain | |

Child tables: feb\_booking,

jan\_booking

Access method: heap

Create Index:

nano=# create index booking\_jan\_idx on jan\_booking using btree(booking\_date);

CREATE INDEX

nano=# create index booking\_feb\_idx on feb\_booking using btree(booking\_date);

CREATE INDEX

Create Function:

create or replace function on\_insert() returns trigger as $$

begin

if(new.booking\_date >= date '2020-01-01' and new.booking\_date <=date '2020-01-31') then

insert into jan\_booking values(new.\*);

elsif (new.booking\_date >= date '2020-02-01' and new.booking\_date <=date '2020-02-29') then

insert into feb\_booking values(new.\*);

else

raise exception 'Enter valid booking date';

end if;

return null;

end;

$$ LANGUAGE plpgsql;

Create Trigger:

create trigger booking\_entry before insert on bookings for each row execute procedure on\_insert();

Insert Records:

nano=# insert into bookings values('dxb102','emirates','2020-02-09');

INSERT 0 0

nano=# insert into bookings values('dxb103','emirates','2020-02-15');

INSERT 0 0

nano=# insert into bookings values('auh345','etihad','2020-01-10');

INSERT 0 0

nano=# select \* from bookings;

flightno | flightname | booking\_date

----------+------------+---------------------

dxb101 | etihad | 2020-01-25 00:00:00

auh345 | etihad | 2020-01-10 00:00:00

dxb102 | emirates | 2020-02-09 00:00:00

dxb103 | emirates | 2020-02-15 00:00:00

(4 rows)

nano=# select \* from only bookings;

flightno | flightname | booking\_date

----------+------------+--------------

(0 rows)

nano=# select \* from jan\_bookings;

ERROR: relation "jan\_bookings" does not exist

LINE 1: select \* from jan\_bookings;

^

nano=# select \* from jan\_booking;

flightno | flightname | booking\_date

----------+------------+---------------------

dxb101 | etihad | 2020-01-25 00:00:00

auh345 | etihad | 2020-01-10 00:00:00

(2 rows)

nano=# select \* from feb\_booking;

flightno | flightname | booking\_date

----------+------------+---------------------

dxb102 | emirates | 2020-02-09 00:00:00

dxb103 | emirates | 2020-02-15 00:00:00

(2 rows)

Validate Function with wrong entry:

nano=# insert into bookings values('auh234','etihad','2020-03-12');

ERROR: Enter valid booking date

CONTEXT: PL/pgSQL function on\_insert() line 8 at RAISE