EXPERIMENT NO.4

AIM: To implement database partitioning on tables

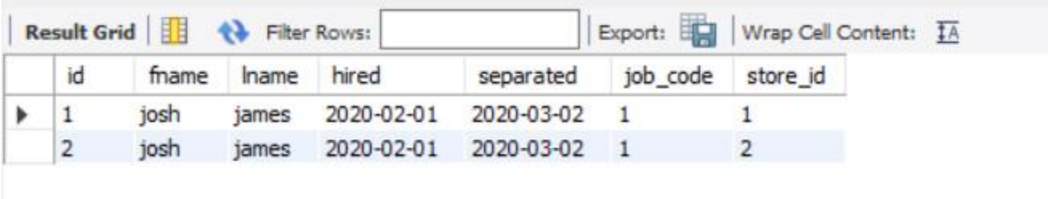
Theory: Partitioning enables users to distribute portions of individual tables across a file system according to rules which you can set largely as needed. In effect, different portions of a table are stored as separate tables in different locations. The function is selected according to the partitioning type specified by the user, and takes as its parameter the value of a user-supplied expression. This expression can be a column value, a function acting on one or more column values, or a set of one or more column values, depending on the type of partitioning that is used.

Types of partitions:

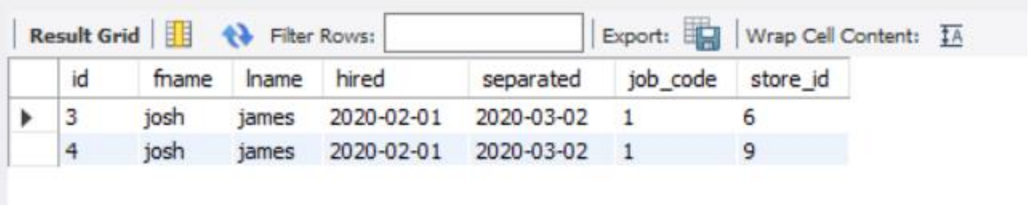
1) RANGE partitioning: This type of partitioning assigns rows to partitions based on column values falling within a given range. A table that is partitioned by range is partitioned in such a way that each partition contains rows for which the partitioning expression value lies within a given range. Ranges should be contiguous but not overlapping, and are defined using the VALUES LESS THAN operator.

1)**RANGE**

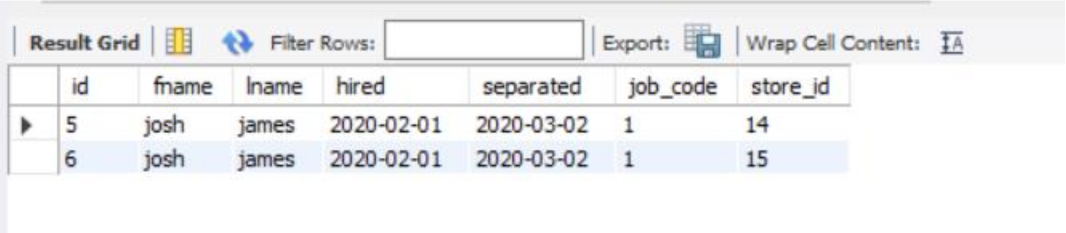
```
CREATE DATABASE EXP1;

USE EXP1;

CREATE TABLE employees (

id INT NOT NULL,fname VARCHAR(30),

lname VARCHAR(30),hired DATE NOT NULL DEFAULT '1970-01-01',

separated DATE NOT NULL DEFAULT '9999-12-31',

job_code INT NOT NULL,store_id INT NOT NULL);


DROP TABLE employees;

CREATE TABLE employees(
id INT NOT NULL,fname VARCHAR(30),lname VARCHAR(30),hired DATE NOT NULL DEFAULT '1970-01-01',

separated DATE NOT NULL DEFAULT '9999-12-31',job_code INT NOT NULL,store_id INT NOT NULL)


PARTITION BY RANGE(store_id)(

PARTITION p0 VALUES LESS THAN(6),

PARTITION p1 VALUES LESS THAN(11),

PARTITION p2 VALUES LESS THAN(16),

PARTITION p3 VALUES LESS THAN(21));


insert into employees values (1, 'josh','james','2020-02-01','2020-03-02',1,1);

insert into employees values (2, 'josh','james','2020-02-01','2020-03-02',1,2);

insert into employees values (3, 'josh','james','2020-02-01','2020-03-02',1,6);

insert into employees values (4, 'josh','james','2020-02-01','2020-03-02',1,9);

insert into employees values (5, 'josh','james','2020-02-01','2020-03-02',1,14);

insert into employees values (6, 'josh','james','2020-02-01','2020-03-02',1,15);

insert into employees values (7, 'josh','james','2020-02-01','2020-03-02',1,17);

insert into employees values (8, 'josh','james','2020-02-01','2020-03-02',1,19);


select * from employees partition(p0);

select * from employees partition(p1);

select * from employees partition(p2);

select * from employees partition(p3);
select count(*) from employees;
```



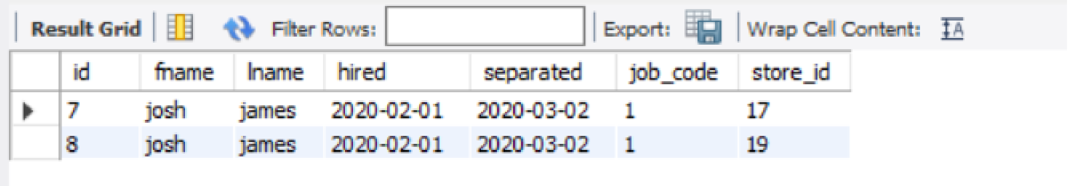
id	fname	lname	hired	separated	job_code	store_id
1	josh	james	2020-02-01	2020-03-02	1	1
2	josh	james	2020-02-01	2020-03-02	1	2



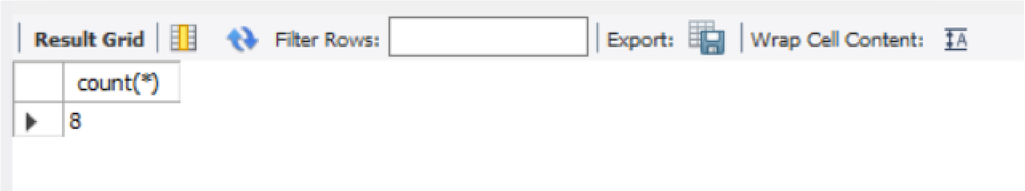
id	fname	lname	hired	separated	job_code	store_id
3	josh	james	2020-02-01	2020-03-02	1	6
4	josh	james	2020-02-01	2020-03-02	1	9



id	fname	lname	hired	separated	job_code	store_id
5	josh	james	2020-02-01	2020-03-02	1	14
6	josh	james	2020-02-01	2020-03-02	1	15



id	fname	lname	hired	separated	job_code	store_id
7	josh	james	2020-02-01	2020-03-02	1	17
8	josh	james	2020-02-01	2020-03-02	1	19

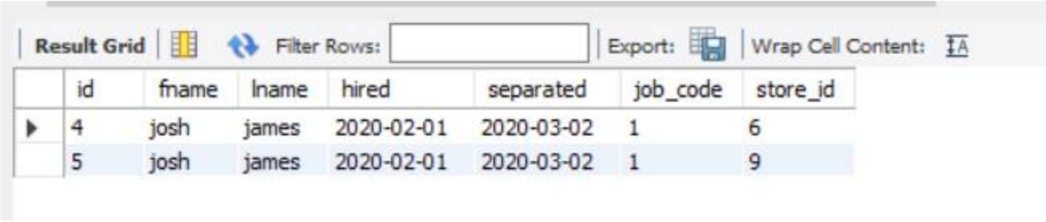


count(*)
8

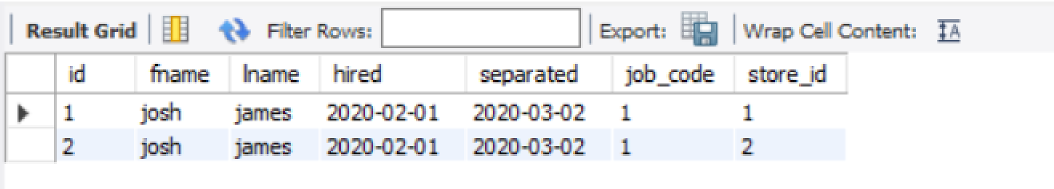
2) **LIST** partitioning: Similar to partitioning by RANGE, except that the partition is selected based on columns matching one of a set of discrete values. This is done by using PARTITION BY LIST(expr) where expr is a column value or an expression based on a column value and returning an integer value, and then defining each partition by means of VALUES IN (value_list), where value_list is a comma-separated list of integers.

Example:

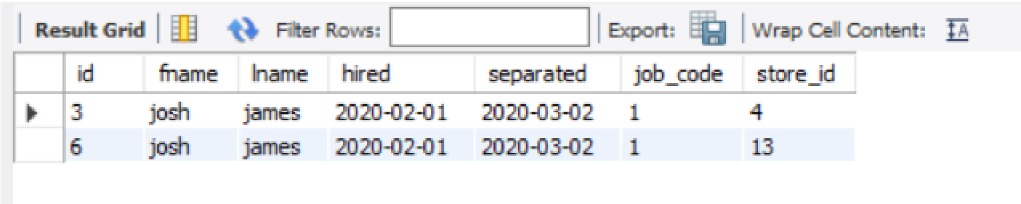
```
DROP TABLE employees;
CREATE TABLE employees(
id INT NOT NULL,
fname VARCHAR(30),
lname VARCHAR(30),
hired DATE NOT NULL DEFAULT'1970-01-01',
separated DATE NOT NULL DEFAULT'9999-12-31',
job_code INT,
store_id INT)
PARTITION BY LIST(store_id)(
PARTITION pNorth VALUES IN(3,5,6,9,17),
PARTITION pEast VALUES IN(1,2,10,11,19,20),
PARTITION pWest VALUES IN(4,12,13,14,18),
PARTITION pCentral VALUES IN(7,8,15,16));

insert into employees values (1, 'josh','james','2020-02-01','2020-03-02',1,1);
insert into employees values (2, 'josh','james','2020-02-01','2020-03-02',1,2);
insert into employees values (3, 'josh','james','2020-02-01','2020-03-02',1,4);
insert into employees values (4, 'josh','james','2020-02-01','2020-03-02',1,6);
insert into employees values (5, 'josh','james','2020-02-01','2020-03-02',1,9);
insert into employees values (6, 'josh','james','2020-02-01','2020-03-02',1,13);
insert into employees values (7, 'josh','james','2020-02-01','2020-03-02',1,15);
insert into employees values (8, 'josh','james','2020-02-01','2020-03-02',1,16);

select * from employees partition(pNorth);
select * from employees partition(pEast);
select * from employees partition(pWest);
select * from employees partition(pCentral);
select count(*) from employees;
```



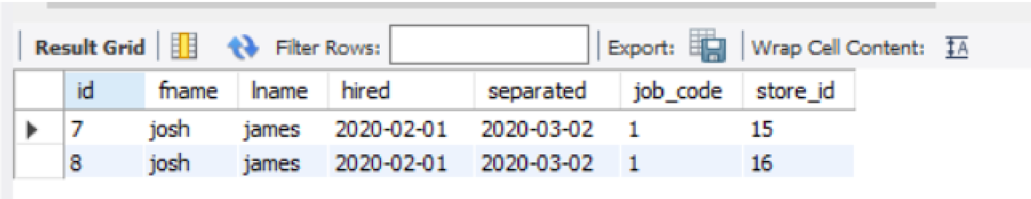
id	fname	lname	hired	separated	job_code	store_id
4	josh	james	2020-02-01	2020-03-02	1	6
5	josh	james	2020-02-01	2020-03-02	1	9



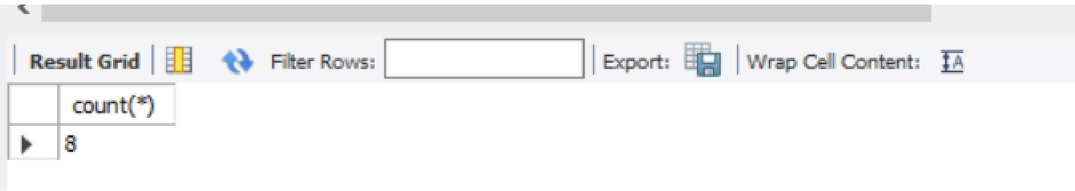
id	fname	lname	hired	separated	job_code	store_id
1	josh	james	2020-02-01	2020-03-02	1	1
2	josh	james	2020-02-01	2020-03-02	1	2



id	fname	lname	hired	separated	job_code	store_id
3	josh	james	2020-02-01	2020-03-02	1	4
6	josh	james	2020-02-01	2020-03-02	1	13



id	fname	lname	hired	separated	job_code	store_id
7	josh	james	2020-02-01	2020-03-02	1	15
8	josh	james	2020-02-01	2020-03-02	1	16

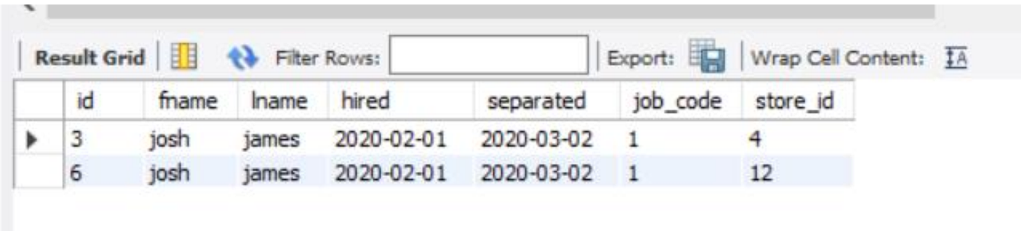


count(*)
8

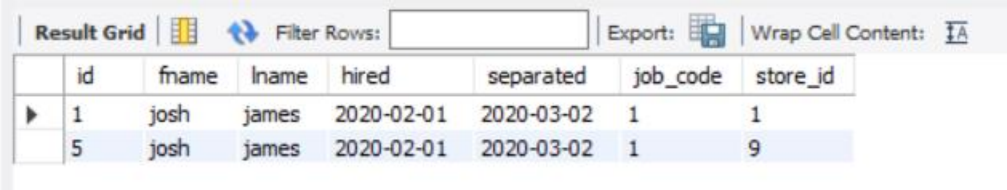
3) HASH partitioning. With this type of partitioning, a partition is selected based on the value returned by a user-defined expression that operates on column values in rows to be inserted into the table. Partitioning by HASH is used primarily to ensure an even distribution of data among a predetermined number of partitions.

Example:

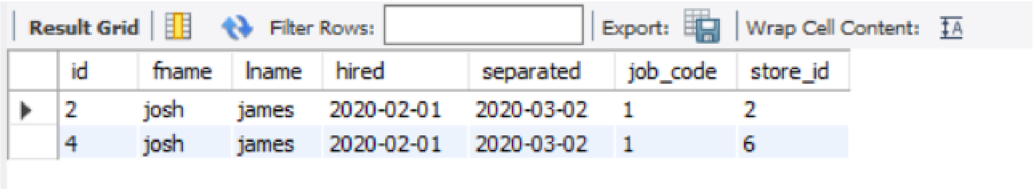
```
DROP TABLE employees;
CREATE TABLE employees(
id INT NOT NULL,
fname VARCHAR(30),
lname VARCHAR(30),
hired DATE NOT NULL DEFAULT '1970-01-01',
separated DATE NOT NULL DEFAULT '9999-12-31',
job_code INT,
store_id INT)
PARTITION BY HASH(store_id) PARTITIONS 4;

insert into employees values (1, 'josh','james','2020-02-01','2020-03-02',1,1);
insert into employees values (2, 'josh','james','2020-02-01','2020-03-02',1,2);
insert into employees values (3, 'josh','james','2020-02-01','2020-03-02',1,4);
insert into employees values (4, 'josh','james','2020-02-01','2020-03-02',1,6);
insert into employees values (5, 'josh','james','2020-02-01','2020-03-02',1,9);
insert into employees values (6, 'josh','james','2020-02-01','2020-03-02',1,12);
insert into employees values (7, 'josh','james','2020-02-01','2020-03-02',1,15);
insert into employees values (8, 'josh','james','2020-02-01','2020-03-02',1,19);

select * from employees partition(p0);
select * from employees partition(p1);
select * from employees partition(p2);
select * from employees partition(p3);

select count(*) from employees;
```



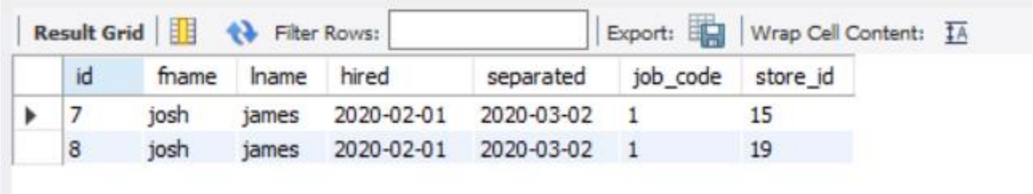
id	fname	lname	hired	separated	job_code	store_id
3	josh	james	2020-02-01	2020-03-02	1	4
6	josh	james	2020-02-01	2020-03-02	1	12



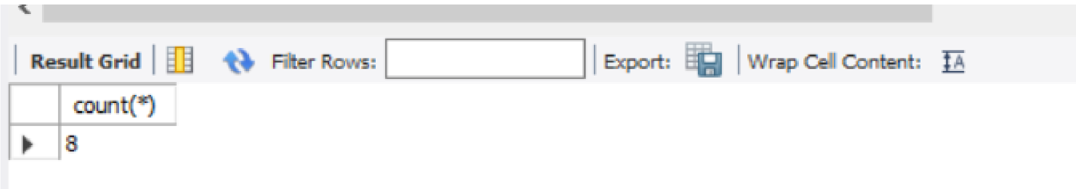
id	fname	lname	hired	separated	job_code	store_id
1	josh	james	2020-02-01	2020-03-02	1	1
5	josh	james	2020-02-01	2020-03-02	1	9



id	fname	lname	hired	separated	job_code	store_id
2	josh	james	2020-02-01	2020-03-02	1	2
4	josh	james	2020-02-01	2020-03-02	1	6



id	fname	lname	hired	separated	job_code	store_id
7	josh	james	2020-02-01	2020-03-02	1	15
8	josh	james	2020-02-01	2020-03-02	1	19

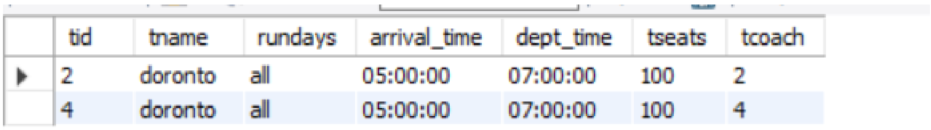


count(*)
8

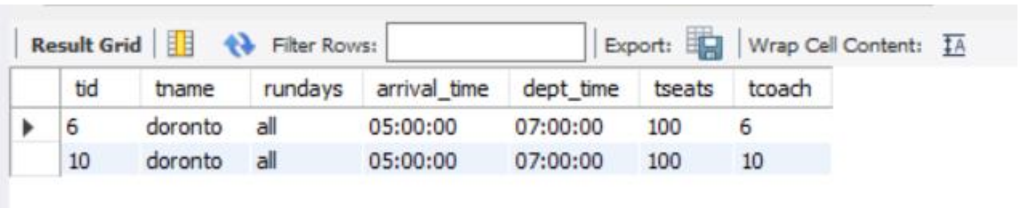
EXERCISE: Create different types of partitions using following table.

1) RANGE

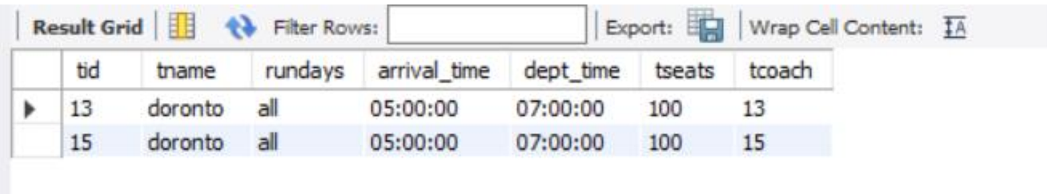
```
CREATE TABLE train2 (
tid INT NOT NULL,
tname VARCHAR (30),
rundays VARCHAR (300),
arrival_time time NOT NULL,
dept_time time NOT NULL,
tseats INT NOT NULL,
tcoach INT NOT NULL)
 PARTITION BY RANGE (tid)
 ( PARTITION p0 VALUES LESS THAN (6),
 PARTITION p1 VALUES LESS THAN (11),
 PARTITION p2 VALUES LESS THAN (16),
 PARTITION p3 VALUES LESS THAN (21) );

 insert into train2 values (2, 'doronto','all','05:00:00','07:00:00',100,2);
 insert into train2 values (4, 'doronto','all','05:00:00','07:00:00',100,4);
 insert into train2 values (6, 'doronto','all','05:00:00','07:00:00',100,6);
 insert into train2 values (10, 'doronto','all','05:00:00','07:00:00',100,10);
 insert into train2 values (13, 'doronto','all','05:00:00','07:00:00',100,13);
 insert into train2 values (15, 'doronto','all','05:00:00','07:00:00',100,15);
 insert into train2 values (18,'doronto','all','05:00:00','07:00:00',100,18);
 insert into train2 values (20, 'doronto','all','05:00:00','07:00:00',100,20);

select * from train2 partition(p0);
select * from train2 partition(p1);
select * from train2 partition(p2);
select * from train2 partition(p3);
select count(*) from train2;
```



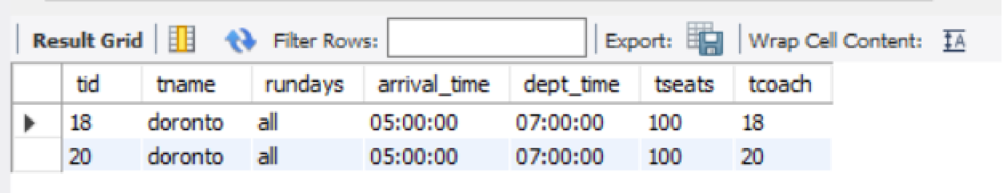
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
2	doronto	all	05:00:00	07:00:00	100	2
4	doronto	all	05:00:00	07:00:00	100	4



tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
6	doronto	all	05:00:00	07:00:00	100	6
10	doronto	all	05:00:00	07:00:00	100	10



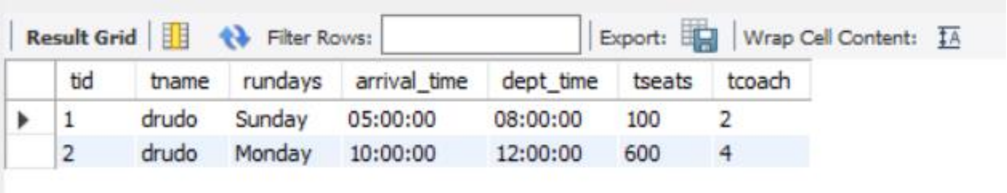
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
13	doronto	all	05:00:00	07:00:00	100	13
15	doronto	all	05:00:00	07:00:00	100	15



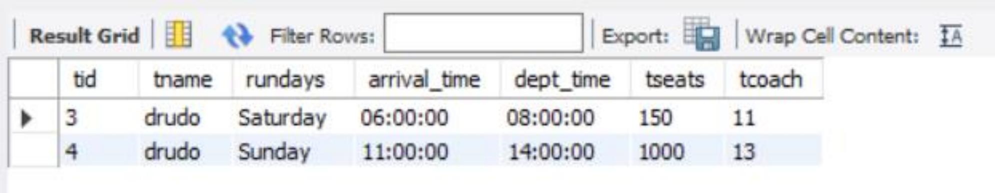
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
18	doronto	all	05:00:00	07:00:00	100	18
20	doronto	all	05:00:00	07:00:00	100	20

2) LIST

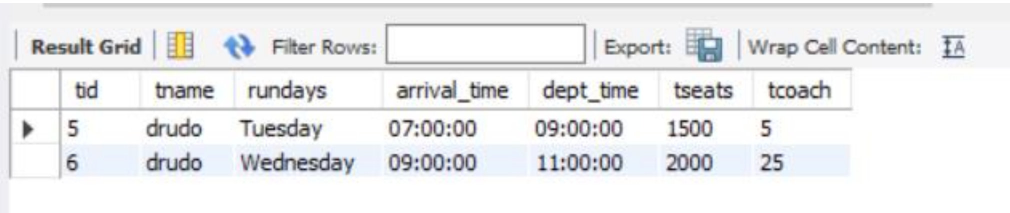
```
DROP TABLE train2;
CREATE TABLE train2(
tid INT NOT NULL,
tname VARCHAR(30),
rundays VARCHAR(300),
arrival_time time NOT NULL,
dept_time time NOT NULL,
tseats INT NOT NULL,
tcoach INT NOT NULL
)
PARTITION BY LIST (tcoach)(
PARTITION upper VALUES IN (1,2,3,4),
PARTITION lower VALUES IN (10,20,11,13),
PARTITION side_upper VALUES IN (6,25,5),
PARTITION side_lower VALUES IN (15,17)
);
insert into train2 values (1, 'drudo','Sunday','05:00:00','08:00:00',100,2);
insert into train2 values (2, 'drudo','Monday','10:00:00','12:00:00',600,4);
insert into train2 values (3, 'drudo','Saturday','06:00:00','08:00:00',150,11);
insert into train2 values (4, 'drudo','Sunday','11:00:00','14:00:00',1000,13);
insert into train2 values (5, 'drudo','Tuesday','07:00:00','09:00:00',1500,5);
insert into train2 values (6, 'drudo','Wednesday','09:00:00','11:00:00',2000,25);
insert into train2 values (7, 'drudo','Monday','10:00:00','12:00:00',600,15);
insert into train2 values (8, 'drudo','Monday','10:00:00','12:00:00',600,17);

select * from train2 partition(upper);
select * from train2 partition(lower);
select * from train2 partition(side_upper);
select * from train2 partition(side_lower);
select count(*) from train2;
```



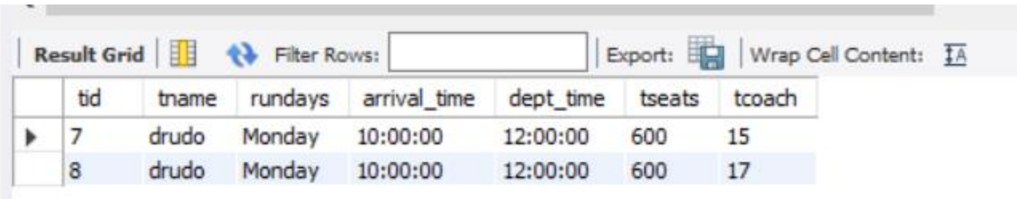
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
1	drudo	Sunday	05:00:00	08:00:00	100	2
2	drudo	Monday	10:00:00	12:00:00	600	4



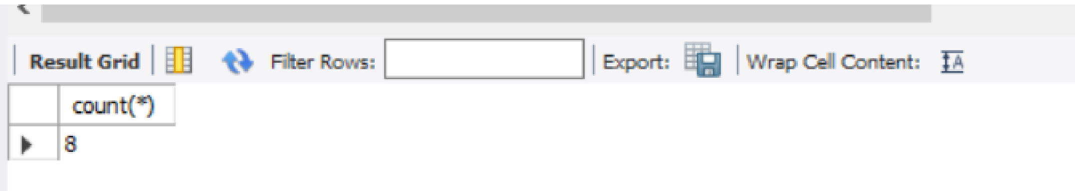
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
3	drudo	Saturday	06:00:00	08:00:00	150	11
4	drudo	Sunday	11:00:00	14:00:00	1000	13



tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
5	drudo	Tuesday	07:00:00	09:00:00	1500	5
6	drudo	Wednesday	09:00:00	11:00:00	2000	25



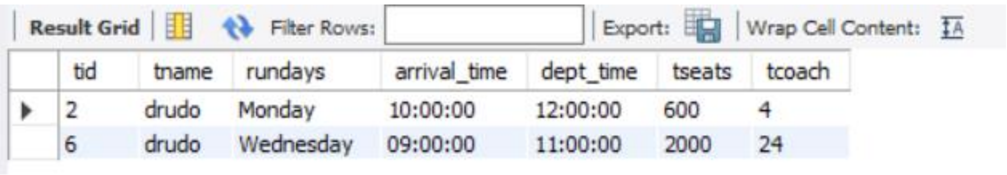
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
7	drudo	Monday	10:00:00	12:00:00	600	15
8	drudo	Monday	10:00:00	12:00:00	600	17



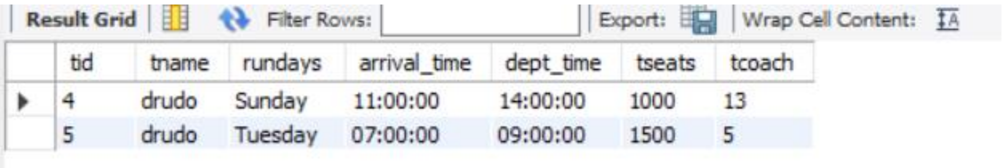
count(*)
8

3) HASH

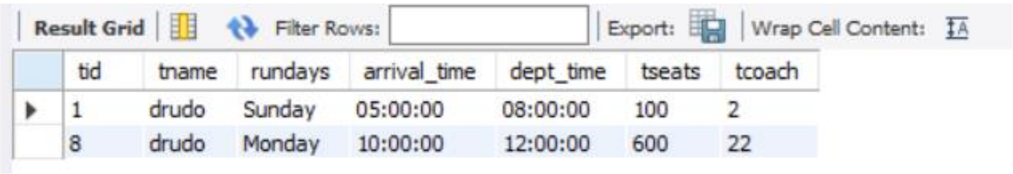
```
DROP TABLE train2;
CREATE TABLE train2(
tid INT NOT NULL,
tname VARCHAR(30),
rundays VARCHAR(300),
arrival_time time NOT NULL,
dept_time time NOT NULL,
tseats INT NOT NULL,
tcoach INT NOT NULL
)
PARTITION BY HASH(tcoach) PARTITIONS 4;
insert into train2 values (1, 'drudo','Sunday','05:00:00','08:00:00',100,2);
insert into train2 values (2, 'drudo','Monday','10:00:00','12:00:00',600,4);
insert into train2 values (3, 'drudo','Saturday','06:00:00','08:00:00',150,11);
insert into train2 values (4, 'drudo','Sunday','11:00:00','14:00:00',1000,13);
insert into train2 values (5, 'drudo','Tuesday','07:00:00','09:00:00',1500,5);
insert into train2 values (6, 'drudo','Wednesday','09:00:00','11:00:00',2000,24);
insert into train2 values (7, 'drudo','Monday','10:00:00','12:00:00',600,7);
insert into train2 values (8, 'drudo','Monday','10:00:00','12:00:00',600,22);

select * from train2 partition(p0);
select * from train2 partition(p1);
select * from train2 partition(p2);
select * from train2 partition(p3);
select count(*) from train2;
```



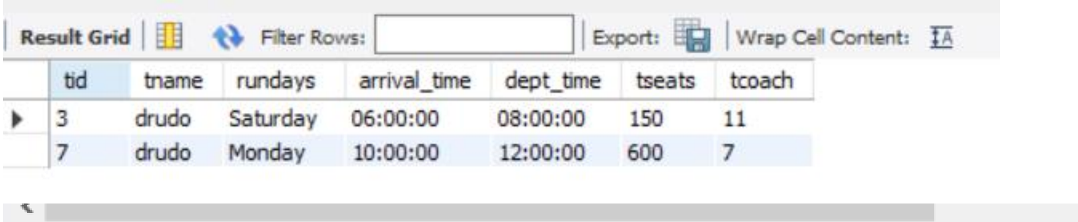
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
2	drudo	Monday	10:00:00	12:00:00	600	4
6	drudo	Wednesday	09:00:00	11:00:00	2000	24



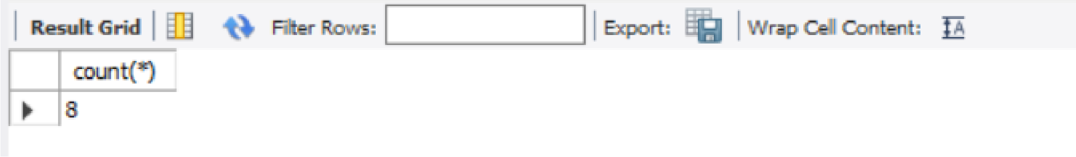
tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
4	drudo	Sunday	11:00:00	14:00:00	1000	13
5	drudo	Tuesday	07:00:00	09:00:00	1500	5



tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
1	drudo	Sunday	05:00:00	08:00:00	100	2
8	drudo	Monday	10:00:00	12:00:00	600	22



tid	tname	rundays	arrival_time	dept_time	tseats	tcoach
3	drudo	Saturday	06:00:00	08:00:00	150	11
7	drudo	Monday	10:00:00	12:00:00	600	7



count(*)
8