Laboratorium V

Wiktor Zmiendak

 Wykonaj partycjonowanie co najmniej 2 tabel zaprojektowanej bazy danych za pomocą RANG, LIST a HASH. Zademonstruj wyniki pracy w postaci zrzutów ekranu skryptów i wyników ich wykonania.

Tabela nr 1 i 2:

```
create table if not exists parking2(
  parking_id int(15) primary key,
  slots_count int(15),
  floor int(15),
  open_hours time,
  cost int(15));

create table if not exists worker2(
    worker_id int(15) primary key,
    worker_name varchar(25),
    age int(15),
    salary int(15),
    working_hours time,
    specialization varchar(25));
```

Uzupełnione dane:

```
insert into airport2.worker2(worker_id, worker_name, age, salary, working_hours, specialization)
values
('1', 'Janek', '60', '3500', '09:00:00', 'worker'),
('2', 'Pawel', '60', '3500', '00:00:00', 'worker'),
('3', 'Michał', '35', '34500', '09:00:00', 'worker'),
('4', 'Piotr', '25', '3500', '09:00:00', 'worker'),
('5', 'Janek', '60', '3500', '09:00:00', 'worker'),
('6', 'Janek', '25', '3500', '09:00:00', 'worker');

insert into airport2.parking2(parking_id, slots_count, floor, open_hours, cost)
values
('1', '100', '3', '09:30:00', '45'),
('2', '200', '2', '00:00:00', '45'),
('3', '1000', '7', '00:00:00', '45'),
('4', '100', '1', '00:00:00', '45'),
('5', '20', '2', '08:00:00', '45'),
('6', '20', '4', '00:00:00', '45');
```

Dokonujemy partycjonowania metodą RANGE:

```
-- RANGE

ALTER TABLE worker2

○ PARTITION BY RANGE (worker_id) (

PARTITION p0 VALUES LESS THAN (2),

PARTITION p1 VALUES LESS THAN (4),

PARTITION p2 VALUES LESS THAN (MAXVALUE)
);

ALTER TABLE parking2

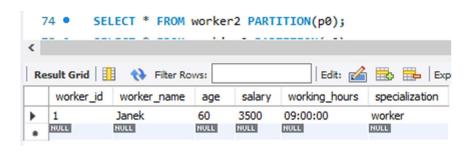
○ PARTITION BY RANGE (parking_id) (

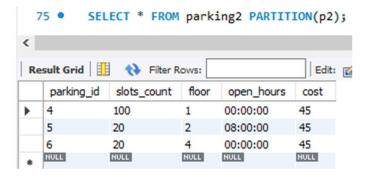
PARTITION p0 VALUES LESS THAN (2),

PARTITION p1 VALUES LESS THAN (4),

PARTITION p2 VALUES LESS THAN (MAXVALUE)
);
```

Wyniki:





Dokonujemy partycjonowanie metodą LIST:

```
-- LIST

ALTER TABLE worker2

PARTITION BY LIST (worker_id) (

PARTITION p0 VALUES IN (1, 2),

PARTITION p1 VALUES IN (3, 4),

PARTITION p2 VALUES IN (5, 6)

);

ALTER TABLE parking2

PARTITION BY LIST (parking_id) (

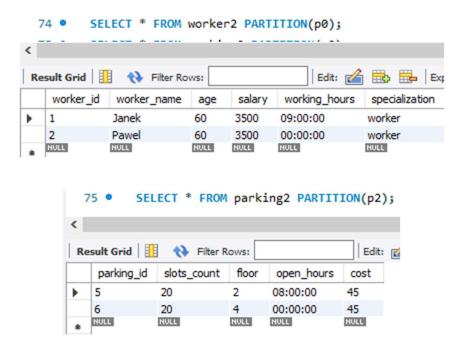
PARTITION p0 VALUES IN (1, 2),

PARTITION p1 VALUES IN (3, 4),

PARTITION p2 VALUES IN (5, 6)

);
```

Wyniki:



Dokonujemy partycjonowanie metodą HASH:

```
-- HASH

ALTER TABLE worker2

PARTITION BY HASH (worker_id) PARTITIONS 4;

ALTER TABLE parking2

PARTITION BY HASH (parking_id) PARTITIONS 4;
```

Wyniki:

