ADBMS Experiment 2

Ojas Patil

2019130048

TE COMPS

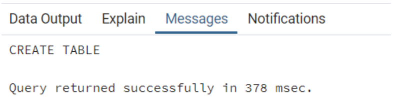
Batch: B

Aim: To design a distributed database by applying the concepts of horizontal fragmentation.

Procedure:

1) Create a global schema

```
-- Create global schema
CREATE TABLE  CUSTOMER
  (       "CUST_CODE" VARCHAR(6) NOT NULL PRIMARY KEY,
                  "CUST_NAME" VARCHAR(40) NOT NULL,
                  "CUST_CITY" CHAR(35),
                  "WORKING_AREA" VARCHAR(35) NOT NULL,
                  "CUST_COUNTRY" VARCHAR(20) NOT NULL,
                  "PAYMENT_AMT" DECIMAL(12,2) NOT NULL,
                  "PHONE_NO" VARCHAR(17) NOT NULL
);
```



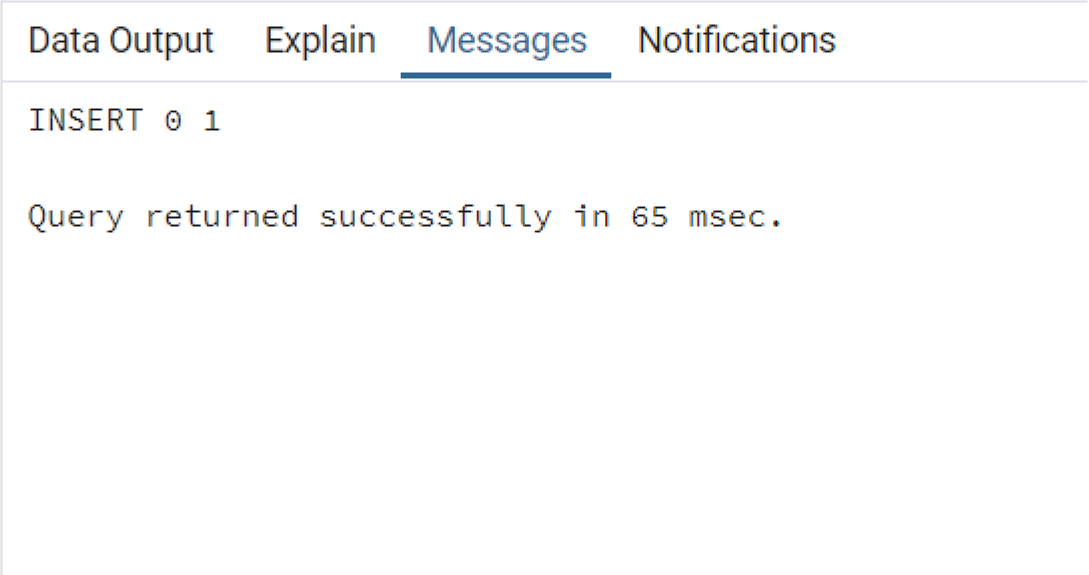
```
CREATE TABLE

Query returned successfully in 378 msec.
```

2) Populate global schema

```
-- Populate global schema
INSERT INTO CUSTOMER VALUES ('C00013', 'Holmes', 'London', 'London', 'UK',
'4000.00', 'BBBBBBB');
INSERT INTO CUSTOMER VALUES ('C00001', 'Micheal', 'New York', 'New York', 'USA',
'6000.00', 'CCCCCCC');
INSERT INTO CUSTOMER VALUES ('C00020', 'Albert', 'New York', 'New York', 'USA',
'6000.00', 'BBBBSBB');
INSERT INTO CUSTOMER VALUES ('C00025', 'Ravindran', 'Bangalore', 'Bangalore',
'India','8000.00', 'AVAVAVA');
INSERT INTO CUSTOMER VALUES ('C00024', 'Cook', 'London', 'London', 'UK',
'6000.00', 'FSDDSDF');
```

```
INSERT INTO CUSTOMER VALUES ('C00015', 'Stuart', 'London', 'London', 'UK', '11000.00', 'GFSGERS');
INSERT INTO CUSTOMER VALUES ('C00002', 'Bolt', 'New York', 'New York', 'USA', '3000.00', 'DDNRDRH');
INSERT INTO CUSTOMER VALUES ('C00018', 'Fleming', 'Brisban', 'Brisban', 'Australia', '5000.00', 'NHBGVFC');
INSERT INTO CUSTOMER VALUES ('C00021', 'Jacks', 'Brisban', 'Brisban', 'Australia', '7000.00', 'WERTGDF');
INSERT INTO CUSTOMER VALUES ('C00019', 'Yearannaidu', 'Chennai', 'Chennai', 'India', '8000.00', 'ZZZZBFV');
INSERT INTO CUSTOMER VALUES ('C00005', 'Sasikant', 'Mumbai', 'Mumbai', 'India', '11000.00', '147-25896312');
INSERT INTO CUSTOMER VALUES ('C00007', 'Ramanathan', 'Chennai', 'Chennai', 'India', '9000.00', 'GHRDWSD');
INSERT INTO CUSTOMER VALUES ('C00022', 'Avinash', 'Mumbai', 'Mumbai', 'India', '9000.00', '113-12345678');
INSERT INTO CUSTOMER VALUES ('C00004', 'Winston', 'Brisban', 'Brisban', 'Australia', '6000.00', 'AAAAAAA');
```

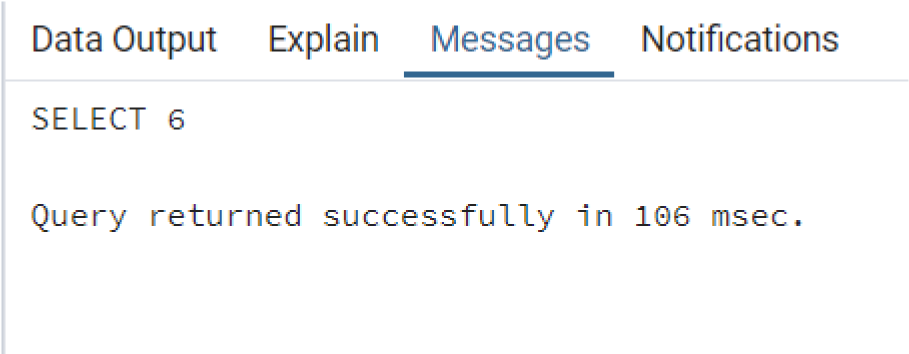


```
INSERT 0 1

Query returned successfully in 65 msec.
```

3) Fragmented table on the basis of payment_amt

```
-- Horizontal fragmentation on the basis of payment_amt of customer table
CREATE TABLE CUSTOMER1 AS SELECT * FROM CUSTOMER WHERE "PAYMENT_AMT" <= 7000;
CREATE TABLE CUSTOMER2 AS SELECT * FROM CUSTOMER WHERE "PAYMENT_AMT" > 7000;
```



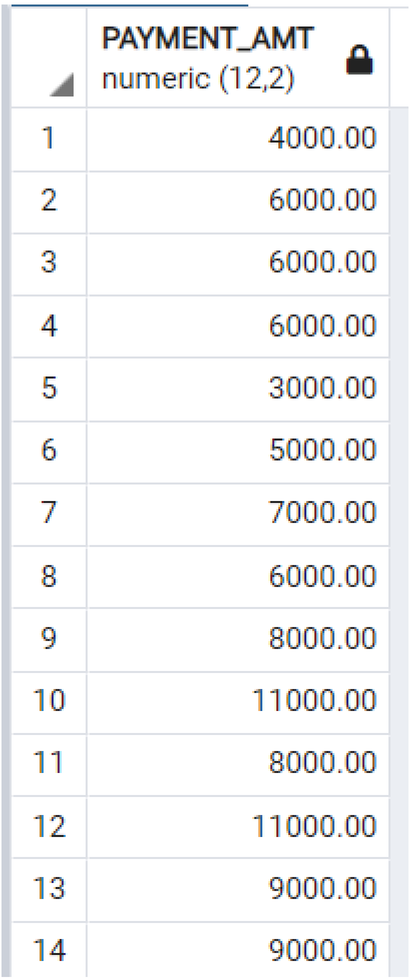
```
SELECT 6

Query returned successfully in 106 msec.
```

4) Answer to Q1

-- Q.1 Finding PAYMENT_AMT of each customer

SELECT "PAYMENT_AMT" FROM CUSTOMER1 UNION ALL SELECT "PAYMENT_AMT" FROM CUSTOMER2;



	PAYMENT_AMT numeric (12,2)
1	4000.00
2	6000.00
3	6000.00
4	6000.00
5	3000.00
6	5000.00
7	7000.00
8	6000.00
9	8000.00
10	11000.00
11	8000.00
12	11000.00
13	9000.00
14	9000.00

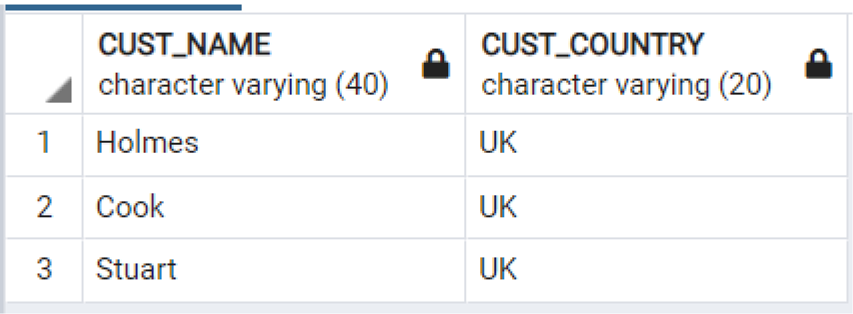
5) Answer to Q2

-- Q.2 Finding Name of customer where customer country is 'UK'

SELECT "CUST_NAME", "CUST_COUNTRY" FROM CUSTOMER1 WHERE "CUST_COUNTRY"='UK'

UNION ALL

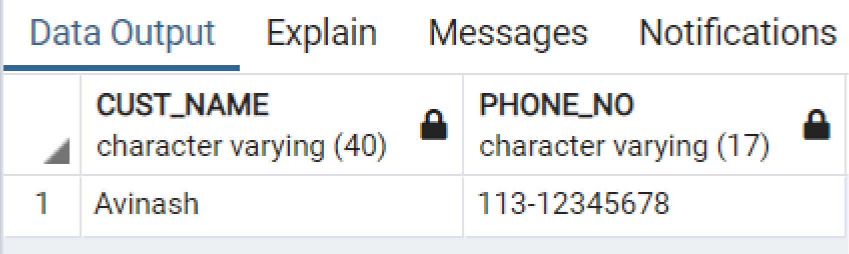
SELECT "CUST_NAME", "CUST_COUNTRY" FROM CUSTOMER2 WHERE "CUST_COUNTRY"='UK';



	CUST_NAME character varying (40)	CUST_COUNTRY character varying (20)
1	Holmes	UK
2	Cook	UK
3	Stuart	UK

6) Answer to Q3

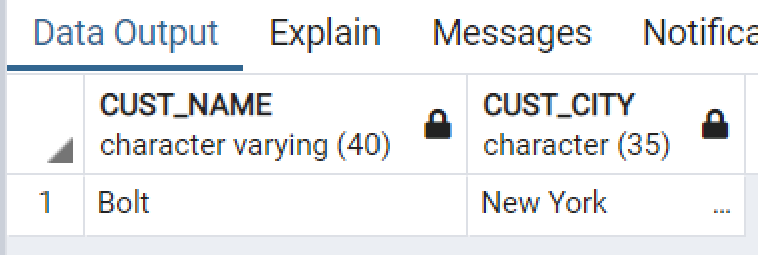
```
-- Q.3 Find customer name and his phone no from customer code
SELECT "CUST_NAME", "PHONE_NO" FROM
(SELECT * FROM CUSTOMER1
 UNION
 SELECT * FROM CUSTOMER2) AS foo WHERE "CUST_CODE" = 'C00022';
```



	CUST_NAME character varying (40)	PHONE_NO character varying (17)
1	Avinash	113-12345678

7) Answer to Q4

```
-- Q.4 Find customer name and city from customer code
SELECT "CUST_NAME", "CUST_CITY" FROM
(SELECT * FROM CUSTOMER1
 UNION
 SELECT * FROM CUSTOMER2) AS foo WHERE "CUST_CODE" = 'C00002';
```



	CUST_NAME character varying (40)	CUST_CITY character (35)
1	Bolt	New York

8) Checking Completeness.

1. There exists a customer named Holmes with payment amount = 4000. If it exists in customer 1 then we can say it satisfies comleteness.

2. Similarly if Ravindran exist in customer 2 then we can say it satisfies completeness. Also sum of entries in bothe the fragments must be equal to no. of entries in customer schema

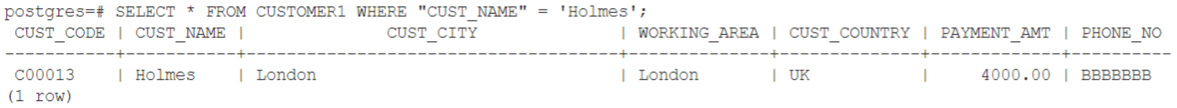
SELECT * FROM CUSTOMER1 WHERE "CUST_NAME" = 'Holmes';

SELECT * FROM CUSTOMER2 WHERE "CUST_NAME" = 'Ravindran';

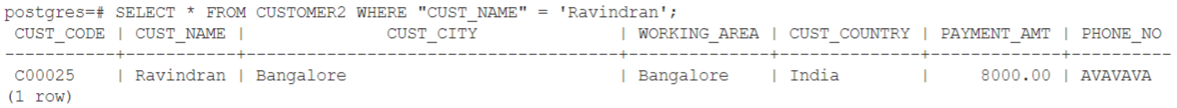
SELECT COUNT(*) FROM CUSTOMER;

SELECT COUNT(*) FROM CUSTOMER1;

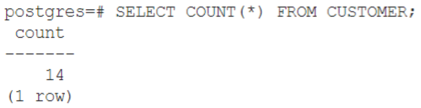
SELECT COUNT(*) FROM CUSTOMER2;



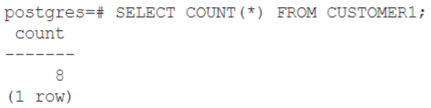
```
postgres=# SELECT * FROM CUSTOMER1 WHERE "CUST_NAME" = 'Holmes';
 CUST_CODE | CUST_NAME |              CUST_CITY                | WORKING_AREA | CUST_COUNTRY | PAYMENT_AMT | PHONE_NO
-----------+-----------+-------------------------------------+--------------+--------------+-------------+----------
 C00013    | Holmes    | London                              | London       | UK           |     4000.00 | BBBBBBB
(1 row)
```



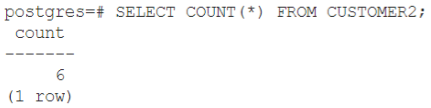
```
postgres=# SELECT * FROM CUSTOMER2 WHERE "CUST_NAME" = 'Ravindran';
 CUST_CODE | CUST_NAME |              CUST_CITY                | WORKING_AREA | CUST_COUNTRY | PAYMENT_AMT | PHONE_NO
-----------+-----------+-------------------------------------+--------------+--------------+-------------+----------
 C00025    | Ravindran | Bangalore                           | Bangalore    | India        |     8000.00 | AVAVAVA
(1 row)
```



```
postgres=# SELECT COUNT(*) FROM CUSTOMER;
 count
-------
    14
(1 row)
```



```
postgres=# SELECT COUNT(*) FROM CUSTOMER1;
 count
-------
     8
(1 row)
```



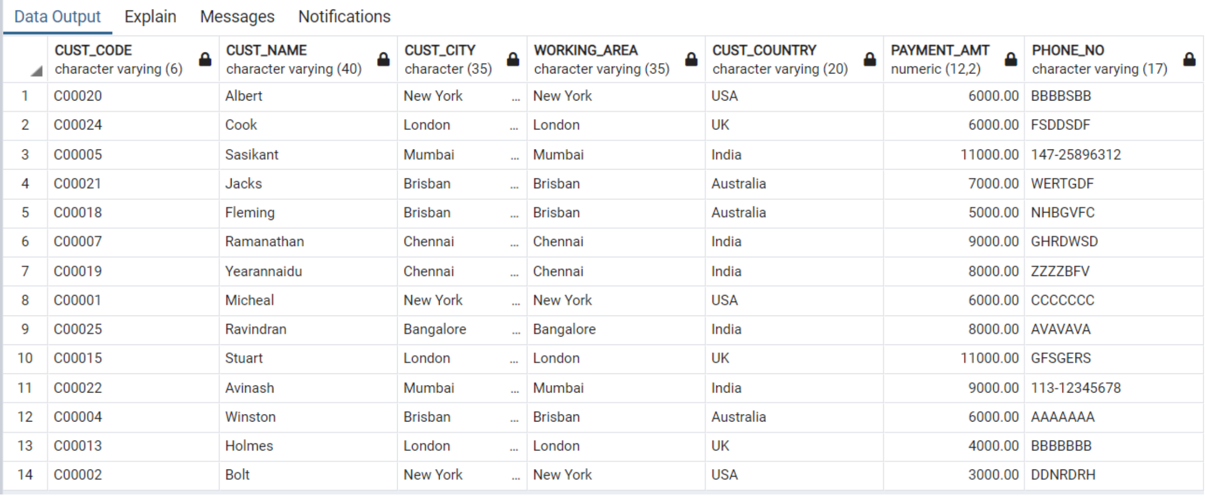
```
postgres=# SELECT COUNT(*) FROM CUSTOMER2;
 count
-------
     6
(1 row)
```

9) Checking reconstruction

SELECT * FROM CUSTOMER1

UNION

SELECT * FROM CUSTOMER2;

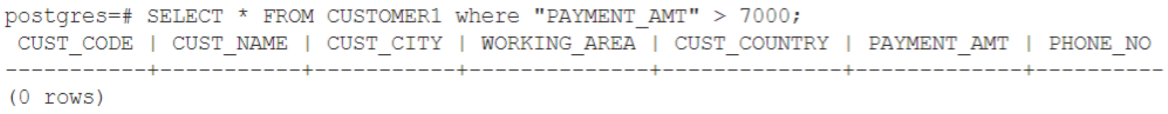


	CUST_CODE character varying (6)	CUST_NAME character varying (40)	CUST_CITY character (35)	WORKING_AREA character varying (35)	CUST_COUNTRY character varying (20)	PAYMENT_AMT numeric (12,2)	PHONE_NO character varying (17)
1	C00020	Albert	New York ...	New York	USA	6000.00	BBBBSBB
2	C00024	Cook	London ...	London	UK	6000.00	FSDDSDF
3	C00005	Sasikant	Mumbai ...	Mumbai	India	11000.00	147-25896312
4	C00021	Jacks	Brisban ...	Brisban	Australia	7000.00	WERTGDF
5	C00018	Fleming	Brisban ...	Brisban	Australia	5000.00	NHBGVFC
6	C00007	Ramanathan	Chennai ...	Chennai	India	9000.00	GHRDWSD
7	C00019	Yearannaidu	Chennai ...	Chennai	India	8000.00	ZZZZBFV
8	C00001	Micheal	New York ...	New York	USA	6000.00	CCCCCCC
9	C00025	Ravindran	Bangalore ...	Bangalore	India	8000.00	AVAVAVA
10	C00015	Stuart	London ...	London	UK	11000.00	GFSGERS
11	C00022	Avinash	Mumbai ...	Mumbai	India	9000.00	113-12345678
12	C00004	Winston	Brisban ...	Brisban	Australia	6000.00	AAAAAAA
13	C00013	Holmes	London ...	London	UK	4000.00	BBBBBBB
14	C00002	Bolt	New York ...	New York	USA	3000.00	DDNRDRH

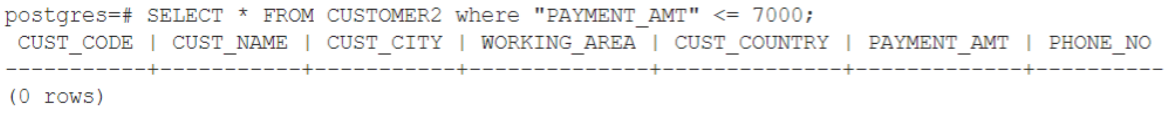
10) Verifying disjointness

On searching any customer with payment amount higher than 7000 in customer 1 we must get an empty set

SELECT * FROM CUSTOMER1 where "PAYMENT_AMT" > 7000;
SELECT * FROM CUSTOMER2 where "PAYMENT_AMT" <= 7000;



```
postgres=# SELECT * FROM CUSTOMER1 where "PAYMENT_AMT" > 7000;
 CUST_CODE | CUST_NAME | CUST_CITY | WORKING_AREA | CUST_COUNTRY | PAYMENT_AMT | PHONE_NO
-----------+-----------+-----------+--------------+--------------+-------------+----------
(0 rows)
```



```
postgres=# SELECT * FROM CUSTOMER2 where "PAYMENT_AMT" <= 7000;
 CUST_CODE | CUST_NAME | CUST_CITY | WORKING_AREA | CUST_COUNTRY | PAYMENT_AMT | PHONE_NO
-----------+-----------+-----------+--------------+--------------+-------------+----------
(0 rows)
```

Conclusion:

From the above experiment, I understood the process of horizontal fragmentation along a primary predicate. In the above experiment, I fragmented global schema based on single attribute’s value. This method gave me two fragments and followed all the correctness rules of fragmented database. I also observed the significant reduction in query execution timings.