For homework 2 solution, query code given below.

a) Partition the Orders table using orderdate with the following constraints:

1. Orders between: 20060703 00:00:00.000 and 20070205 00:00:00.000
2. Orders between: 20070205 00:00:00.000 and 20070819 00:00:00.000
3. Orders between: 20070819 00:00:00.000 and 20080123 00:00:00.000
4. Orders between: 20080123 00:00:00.000 and 20080507 00:00:00.000

Query code-

```
CREATE TABLE Orders
(
  orderid       INT         NOT NULL,
  custid        INT         NULL,
  empid         INT         NOT NULL,
  orderdate     TIMESTAMP     NOT NULL,
  requireddate   TIMESTAMP     NOT NULL,
  shippeddate    TIMESTAMP     NULL,
  shipperid      INT         NOT NULL,
  freight       MONEY       NOT NULL,
  shipname      VARCHAR(40) NOT NULL,
  shipaddress    VARCHAR(60) NOT NULL,
  shipcity      VARCHAR(15) NOT NULL,
  shipregion     VARCHAR(15) NULL,
  shippostalcode VARCHAR(10) NULL,
  shipcountry    VARCHAR(15) NOT NULL
) PARTITION BY RANGE (orderdate);

CREATE TABLE first_partition PARTITION OF Orders
    FOR VALUES FROM ('2006-07-03') TO ('2007-02-05');
CREATE TABLE second_partition PARTITION OF Orders
    FOR VALUES FROM ('2007-02-05') TO ('2007-08-19');
CREATE TABLE third_partition PARTITION OF Orders
    FOR VALUES FROM ('2007-08-19') TO ('2008-01-23 ');
CREATE TABLE fourth_partition PARTITION OF Orders
    FOR VALUES FROM ('2008-01-23') TO ('2008-05-07');
```

b) Alter the third partition and add a contraint where the freight cost is higher than 50 €

Query code-

```
CREATE TABLE third_partition
  (LIKE orders INCLUDING DEFAULTS INCLUDING CONSTRAINTS);

ALTER TABLE third_partition ADD CONSTRAINT freight
   CHECK ( freight > 50);
```

c) Alter the fourth partition and add a constraint that the shipped date should not be null

Query code-

```
CREATE TABLE fourth_partition

  (LIKE orders INCLUDING DEFAULTS INCLUDING CONSTRAINTS);

ALTER TABLE fourth_partition ADD CONSTRAINT shippeddate

   CHECK ( shippeddate IS NOT NULL);
```

e) How many rows are in each partition?

Query code-

```
SET enable_partition_pruning = on;

EXPLAIN SELECT count(*) FROM orders;
```

My answer shows 160 rows per partition.

homework2/postgres@PostgreSQL 14

Query Editor | Query History

```
1
2  SET enable_partition_pruning = on;
3  EXPLAIN SELECT count(*) FROM orders;
4
```

Data Output | Explain | Messages | Notifications

	QUERY PLAN text
1	Aggregate (cost=51.20..51.21 rows=1 width=8)
2	[...] -> Append (cost=0.00..49.60 rows=640 width=0)
3	[...] -> Seq Scan on order_y2006_07 orders_1 (cost=0.00..11.60 rows=160 width=0)
4	[...] -> Seq Scan on order_y2007 orders_2 (cost=0.00..11.60 rows=160 width=0)
5	[...] -> Seq Scan on order_y2008 orders_3 (cost=0.00..11.60 rows=160 width=0)
6	[...] -> Seq Scan on order_y2009 orders_4 (cost=0.00..11.60 rows=160 width=0)