

1)

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'demo@localhost' database structure, including tables like 'departure\_airport', 'arrival\_airport', 'status', 'aircraft\_code', 'actual\_departure', and 'actual\_arrival'. The right pane shows the execution plan for a query. The query is:

```

select airport_name from airports
except
select city from airports
order by airport_name;

```

The output window shows the results of the query, which are 11 rows of airport names:

airport_name
1 Байкал
2 Баратаевка
3 Бегшево
4 Беспан
5 Бесовец
6 Богашево
7 Витязево
8 Внуково
9 Гумрак
10 Домодедово
11 Донское

2)

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'demo@localhost' database structure. The right pane shows the execution plan for a query. The query is:

```

select airport_name from airports
except
select city from airports
order by airport_name;

```

The output window shows the results of the query, which are 11 rows of airport names:

airport_name
1 Абакан
2 Анадырь
3 Астрахань
4 Барнаул
5 Белгород
6 Белоярский
7 Братск
8 Брянск
9 Бугульма
10 Владивосток
11 Воркута

3)

Database Explorer

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo@localhost

- columns 3
  - book\_ref char(6)
  - book\_date timestamp
  - total\_amount number
- keys 1
- indexes 1
- flights
  - columns 10
    - flight\_id integer = next
    - flight\_no char(6)
    - scheduled\_departure

```

53 SELECT city FROM airports
54 ORDER BY airport_name;
55
56 select count(flight_id) from flights f
57 where departure_airport = 'KZN'
58 and scheduled_departure between '2017-08-01' and '2017-09-01'
59 and extract(MONTH from scheduled_departure) = 8
60 group by aircraft_code
61 having count(flight_id) > 50
62 order by count(flight_id) desc, aircraft_code;
63

```

Services

Output count(flight\_id):bigint x

count	aircraft_code
62	
62	
54	

Database Consoles > demo@localhost > console\_1 [demo@localhost]

61:47 CRLF UTF-8 4 spaces 9:06 08.12.2023

4)

Database Explorer

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo@localhost

- columns 3
  - book\_ref char(6)
  - book\_date timestamp
  - total\_amount number
- keys 1
- indexes 1
- flights
  - columns 10
    - flight\_id integer = next
    - flight\_no char(6)
    - scheduled\_departure
    - scheduled\_arrival
    - departure\_airport

```

90 -- FROM tickets
91 -- WHERE passenger_name LIKE 'IRiNa'
92 -- GROUP BY passenger_name;
93
94 select *
95 from airports_data
96 inner join flights f on airports_data.airport_code = f.departure_airport
97 inner join aircrafts_data a on f.aircraft_code = a.aircraft_code
98 inner join seats s on a.aircraft_code = s.aircraft_code
99 inner join ticket_flights tf on f.flight_id = tf.flight_id
100 inner join boarding_passes bp on tf.ticket_no = bp.ticket_no
101 inner join tickets t on tf.ticket_no = t.ticket_no
102 inner join bookings b on t.book_ref = b.book_ref;
103

```

Services

Output Result 47 x

airport_code	airport_name	city	coordinates	t
1 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
2 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
3 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
4 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
5 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
6 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
7 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc
8 SVO	{"en": "Sheremetyevo International Airport", "ru": "Шереметьево"	{"en": "Moscow", "ru": "Москва"}	(37.4146,55.972599)	Eurc

Database Consoles > demo@localhost > console\_1 [demo@localhost]

102:50 CRLF UTF-8 4 spaces 9:47 08.12.2023

5)

Database Explorer

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo@localhost

- columns 3
  - book\_ref char(6)
  - book\_date timestamp
  - total\_amount numeric
- keys 1
- indexes 1
- flights
  - columns 10
    - flight\_id integer = next
    - flight\_no char(6)
    - scheduled\_departure timestamp
    - scheduled\_arrival timestamp
    - departure\_airport char(6)

```

104
105 ✓ select f.flight_no from ticket_flights t
106 right join flights f on f.flight_id = t.flight_id
107 where t.flight_id is null;
108
109
110
111
112

```

Services

TX

- airports 69 ms
- airports 69 ms
- bookings 160 ms
- bookings
- bookings 160 ms
- flights 185 ms
- flights
- flights

Output demo.bookings.flights

flight_no
1 P60440
2 P60403
3 P60402
4 P60402
5 P60403
6 P60402
7 P60402
8 P60403
9 P60402

Database Consoles > demo@localhost > console\_1 [demo@localhost]

105:20 CRLF UTF-8 4 spaces

6)

Database Explorer

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo@localhost

- columns 3
  - book\_ref char(6)
  - book\_date timestamp
  - total\_amount numeric
- keys 1
- indexes 1
- flights
  - columns 10
    - flight\_id integer = next
    - flight\_no char(6)
    - scheduled\_departure timestamp

```

65
66
67 ✓ SELECT t.ticket_no, SUM(t.amount) AS total_revenue
68 FROM ticket_flights t
69 JOIN flights f ON t.flight_id = f.flight_id
70 WHERE t.ticket_no is not null
71 AND scheduled_departure between '2017-08-01' and '2017-09-01'
72 and extract(MONTH from scheduled_departure) = 8
73 GROUP BY t.ticket_no
74 ORDER BY total_revenue DESC;
75

```

Services

TX

- airports 69 ms
- airports 69 ms
- bookings 160 ms
- bookings
- bookings 160 ms
- flights 185 ms
- flights
- flights
- flights 185 ms
- console\_1 4 s 646 ms
- console\_1 4 s

Output Result 40

ticket_no	total_revenue
1 0005434731223	480600
2 0005434731179	444300
3 0005432664161	433200
4 0005435856241	431800
5 0005435856245	431800
6 0005433461591	428600
7 0005432537068	428000
8 0005434731241	427600
9 0005434731200	427600
10 0005432537097	426600
11 0005434731218	424800

Database Consoles > demo@localhost > console\_1 [demo@localhost]

74:29 CRLF UTF-8 4 spaces

7)

Database Explorer

- demo@localhost
  - columns 3
    - book\_ref char(6)
    - book\_date timestamp
    - total\_amount number
  - keys 1
  - indexes 1
  - flights
    - columns 10
      - flight\_id integer = not null
      - flight\_no char(6)
      - scheduled\_departure timestamp
      - scheduled\_arrival timestamp
      - departure\_airport char(6)

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo: bookings, public console\_1

```

75 SELECT
76     f.flight_no AS flight_no,
77     SUM(CASE WHEN extract(MONTH from scheduled_departure) = 8 THEN t.amount ELSE 0 END) AS august,
78     SUM(CASE WHEN extract(MONTH from scheduled_departure) = 9 THEN t.amount ELSE 0 END) AS september
79 FROM ticket_flights t
80 JOIN flights f ON t.flight_id = f.flight_id
81 WHERE t.ticket_no is not null
82 AND extract(YEAR from scheduled_departure) = 2017
83 GROUP BY f.flight_no
84 ORDER BY
85     SUM(CASE WHEN extract(MONTH from scheduled_departure) = 8 THEN t.amount ELSE 0 END) +
86     SUM(CASE WHEN extract(MONTH from scheduled_departure) = 9 THEN t.amount ELSE 0 END) DESC, flight_no;
87
88

```

Services

TX

- airports\_data 69 ms
- airports 69 ms
- tickets
- bookings 160 ms
- bookings 160 ms
- bookings 160 ms
- flights 185 ms
- flights 185 ms
- flights 185 ms

Output Result 43

flight_no	august	september
1 P60209	480332200	89546000
2 P60208	490523100	51378300
3 P60357	331629500	83622200
4 P60223	340201000	63801500
5 P60222	356857100	38532200
6 P60356	307950500	50770600
7 P60278	293283200	53440800
8 P60199	272368800	68006900
9 P60277	294280000	39832800

Database Consoles > demo@localhost > console\_1 [demo@localhost]

87:105 CRLF UTF-8 4 spaces

9:36 08.12.2023

8)

Database Explorer

- demo@localhost
  - columns 3
    - book\_ref char(6)
    - book\_date timestamp
    - total\_amount number
  - keys 1
  - indexes 1
  - flights
    - columns 10
      - flight\_id integer = not null
      - flight\_no char(6)
      - scheduled\_departure timestamp
      - scheduled\_arrival timestamp
      - departure\_airport char(6)

console [demo@localhost] console\_1 [demo@localhost] x demo airports [demo@localhost] airports\_data [demo@localhost] bookings [demo@localhost]

demo: bookings, public console\_1

```

109 select passenger_name
110 from tickets
111 where passenger_name ilike '%IRiNa%'
112 group by passenger_name;
113
114
115

```

Services

TX

- console
- airports\_data 245 ms
- airports\_data 245 ms
- aircrafts\_data
- airports 69 ms
- airports 69 ms
- tickets
- bookings 160 ms
- bookings 160 ms
- bookings 160 ms
- flights 185 ms
- flights 185 ms
- flights 185 ms
- console\_1 1 s 72 ms
- console\_1 1 s 72 ms

Output demo.bookings.tickets

passenger_name
1 IRINA NOVIKOVA
2 IRINA MIKHAYLOVA
3 IRINA FILIPPOVA
4 IRINA MOROZOVA
5 IRINA ERMAKOVA
6 IRINA MEDVEDEVA
7 IRINA ABRAMOVA
8 IRINA MOISEEVA
9 IRINA KAZAKOVA
10 IRINA ORLOVA
11 IRINA LUKYANOVA
12 IRINA NIKITINA
13 IRINA YAKOVLEVA
14 IRINA STEPANOVA
15 IRINA AFANASEVA

Database Consoles > demo@localhost > console\_1 [demo@localhost]

113:1 CRLF UTF-8 4 spaces

9:50 08.12.2023