

1)

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'demo@localhost' database structure, including tables like 'airports', 'bookings', and 'flights'. The central pane shows a query execution plan for a SELECT statement. The right pane displays the results of the query, which is a list of airport names.

Query:

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

Results:

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 central pane shows a query execution plan for an INTERSECT query. The right pane displays the results of the query, which is a list of airport names.

Query:

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

SELECT airport_name FROM airports
INTERSECT
SELECT city FROM airports
ORDER BY airport_name;
```

Results:

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] console_1

```

5 SELECT (airport_code || airport_name || city || coordinates || timezone) AS full_information
6 FROM airports_data
7 ORDER BY full_information;
8
9 SELECT airport_name, count(departure_airport) from airports_data a, flights f
10 where f.departure_airport = a.airport_code and a.airport_code in ('KZN', 'DME', 'OVB', 'IKT', 'LED', 'SVO')
11 group by airport_name
12 order by count(departure_airport) desc;

```

Services

Output Result 5

airport_name	count
{ "en": "Domodedovo International Airport", "ru": "Домодедово" }	6376
{ "en": "Sheremetyevo International Airport", "ru": "Шереметьево" }	5912
{ "en": "Pulkovo Airport", "ru": "Пулково" }	3769
{ "en": "Tolmachevo Airport", "ru": "Толмачёво" }	2891
{ "en": "Kazan International Airport", "ru": "Казань" }	934
{ "en": "Irkutsk Airport", "ru": "Иркутск" }	727

Database Consoles > demo@localhost > console_1 [demo@localhost]

4)

Database Explorer

console [demo@localhost] console_1 [demo@localhost] x demo airports [demo@localhost] airports_data [demo@localhost] bookings [demo@localhost] console_1

```

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

count
62
62
54

Database Consoles > demo@localhost > console_1 [demo@localhost]

5)

Database Explorer

console [demo@localhost] console_1 [demo@localhost] 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 = new
 - flight_no char(6)
 - scheduled_departure timestamp
 - scheduled_arrival timestamp
 - departure_airport char(6)

```

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

TX

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

Output

airport_code	airport_name	city	coordinates	t
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc
SV0	{ "en": "Sheremetyevo International Airport", "	{ "en": "Moscow", "ru": "Москва" }	(37.4146,55.972599)	Eurc

Database Consoles > demo@localhost > console_1 [demo@localhost]

102:50 CRLF UTF-8 4 spaces

6)

Database Explorer

console [demo@localhost] console_1 [demo@localhost] 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 = new
 - 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
- bookings 160 ms
- flights 185 ms
- flights
- flights

Output

flight_no
P68448
P68403
P68402
P68402
P68403
P68402
P68402
P68403
P68402

Database Consoles > demo@localhost > console_1 [demo@localhost]

105:20 CRLF UTF-8 4 spaces

7)

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

```

65 -- WHERE t.ticket_no IS NULL;
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

Output Result 40 x

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

9:25 08.12.2023

8)

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

```

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

```

Services

Output Result 43 x

flight_no	august	september
1 P60209	480332200	89546000
2 P60208	490523100	51378300
3 P60357	331629500	83622200
4 P60223	348201000	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)

aliyaVersion control

console [demo@localhost]console_1 [demo@localhost]demoairports [demo@localhost]airports_data [demo@localhost]bookings [demo@localhost]

Database Explorer

demo@localhost

- columns 3
 - book_ref char(6)
 - book_date timestamp
 - totalAmount number
- keys 1
- indexes 1
- flights

109 ✓
110
111
112
113
114
115

select passenger_name
from tickets
where passenger_name ilike '%IRiNa%'
group by passenger_name;

Services

Tx:

- console
- airports_data 245 ms
- airports_data 245 ms
- aircrafts_data
- airports 69 ms
- airports 69 ms
- tickets
- bookings 160 ms
- bookings
- bookings 160 ms
- flights 185 ms
- flights
- flights
- flights 185 ms
- console_1 1 s 72 ms
- console_1 1 s 72 ms

Outputdemo.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]

113:1CRLFUTF-84 spaces9:5008.12.2023