

KBTU

FIT

Laboratory work 2: DDL, DML

Kislitsin Alexandr 23B031861

Group: WED 14.00 – 15.00

1. Purpose of the work

The purpose of this laboratory work is to practice creating relational database tables using Data Definition Language (DDL), defining primary keys and foreign keys, normalizing the schema, and performing Data Manipulation Language (DML) operations.

2. Tasks

Create the tables with attributes: Airline, Airport, Passengers, Flights, Booking, Booking_flight, Boarding_pass, Baggage, Baggage_check, Security_check.

Define **Primary Keys** for all tables.

Define **NOT NULL** constraint for all attributes.

Rename table **Airline_info** → **Airline**.

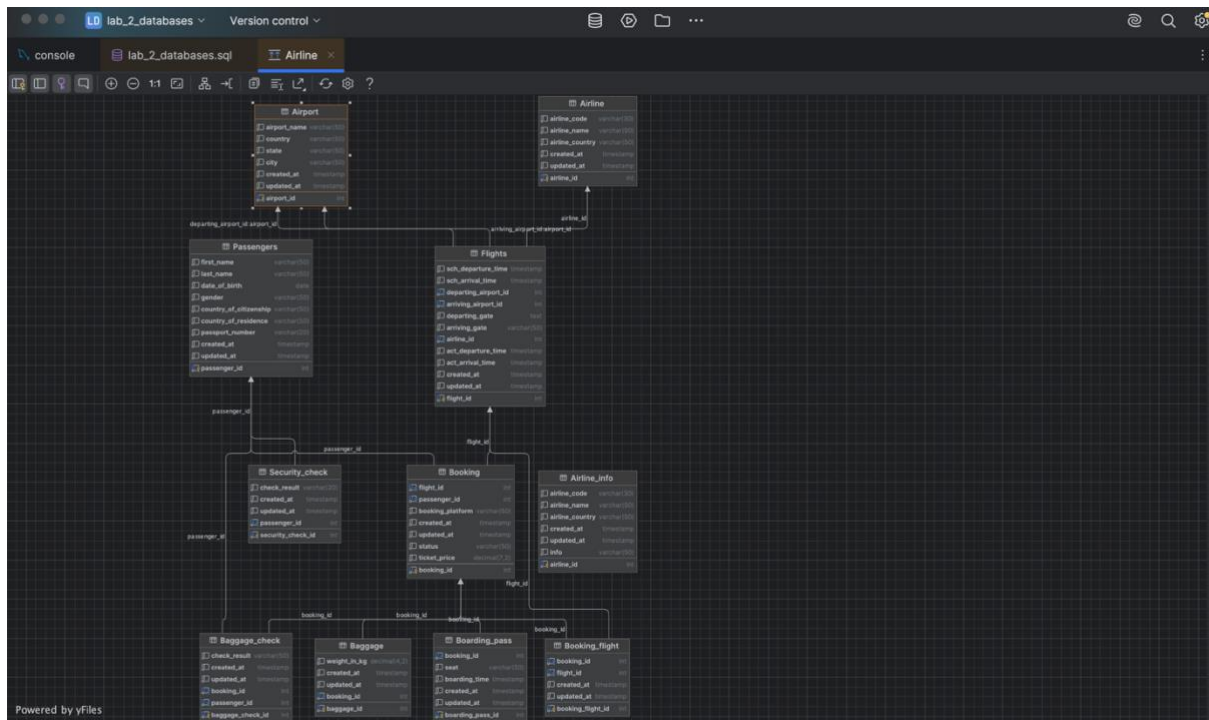
Rename column **price** → **ticket_price** in Booking.

Change column type of **departing_gate** → **TEXT**.

Drop column **info** from Airline.

Define relationships between tables using **FOREIGN KEY** constraints.

Tables



Primary keys for all tables. NOT NULL constraint for all attributes.

```

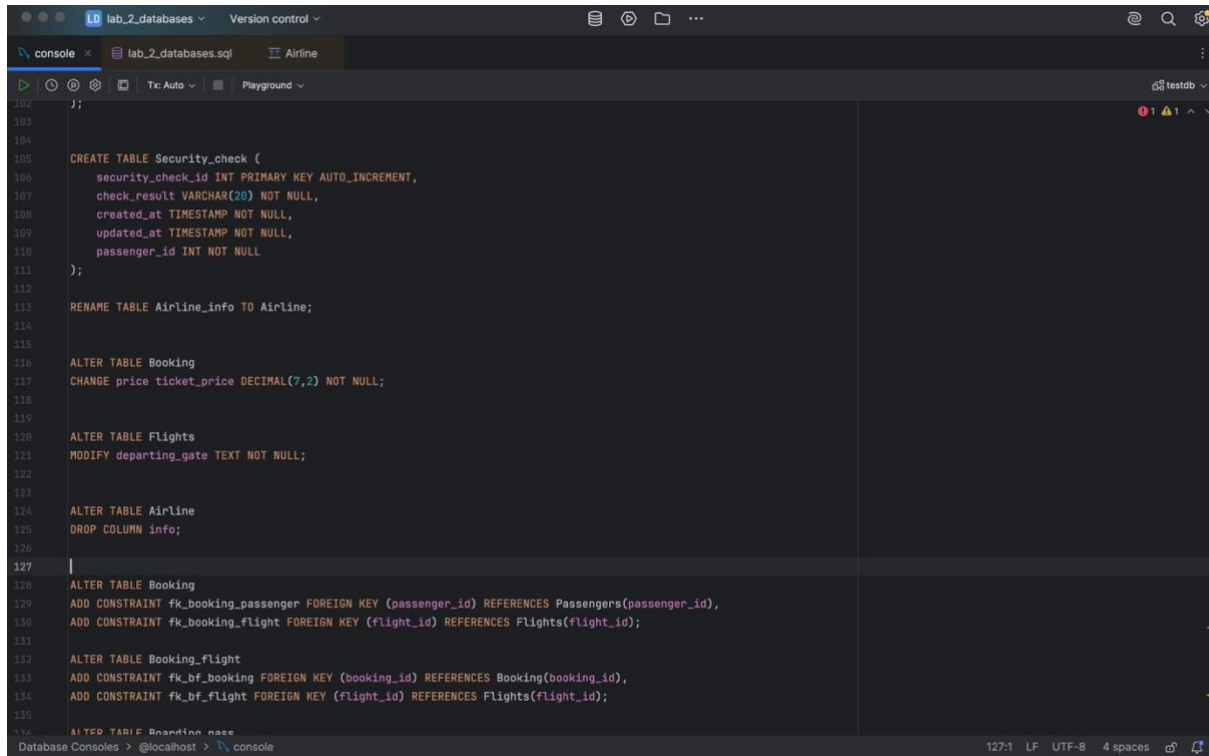
3
4
5 CREATE TABLE Airline_info (
6     airline_id INT PRIMARY KEY AUTO_INCREMENT,
7     airline_code VARCHAR(30) NOT NULL,
8     airline_name VARCHAR(50) NOT NULL,
9     airline_country VARCHAR(50) NOT NULL,
10    created_at TIMESTAMP NOT NULL,
11    updated_at TIMESTAMP NOT NULL,
12    info VARCHAR(50) NOT NULL
13);
14
15 CREATE TABLE Airport (
16     airport_id INT PRIMARY KEY AUTO_INCREMENT,
17     airport_name VARCHAR(50) NOT NULL,
18     country VARCHAR(50) NOT NULL,
19     state VARCHAR(50) NOT NULL,
20     city VARCHAR(50) NOT NULL,
21     created_at TIMESTAMP NOT NULL,
22     updated_at TIMESTAMP NOT NULL
23);
24
25 CREATE TABLE Passengers (
26     passenger_id INT PRIMARY KEY AUTO_INCREMENT,
27     first_name VARCHAR(50) NOT NULL,
28     last_name VARCHAR(50) NOT NULL,
29     date_of_birth DATE NOT NULL,
30     gender VARCHAR(50) NOT NULL,
31     country_of_citizenship VARCHAR(50) NOT NULL,
32     country_of_residence VARCHAR(50) NOT NULL,
33     passport_number VARCHAR(20) NOT NULL UNIQUE,
34     created_at TIMESTAMP NOT NULL,
35     updated_at TIMESTAMP NOT NULL
36
  
```

Rename airline_info table to airline;

Rename column price to ticket_price in booking table;

Change data type of departing_gate from varchar(50) to text;

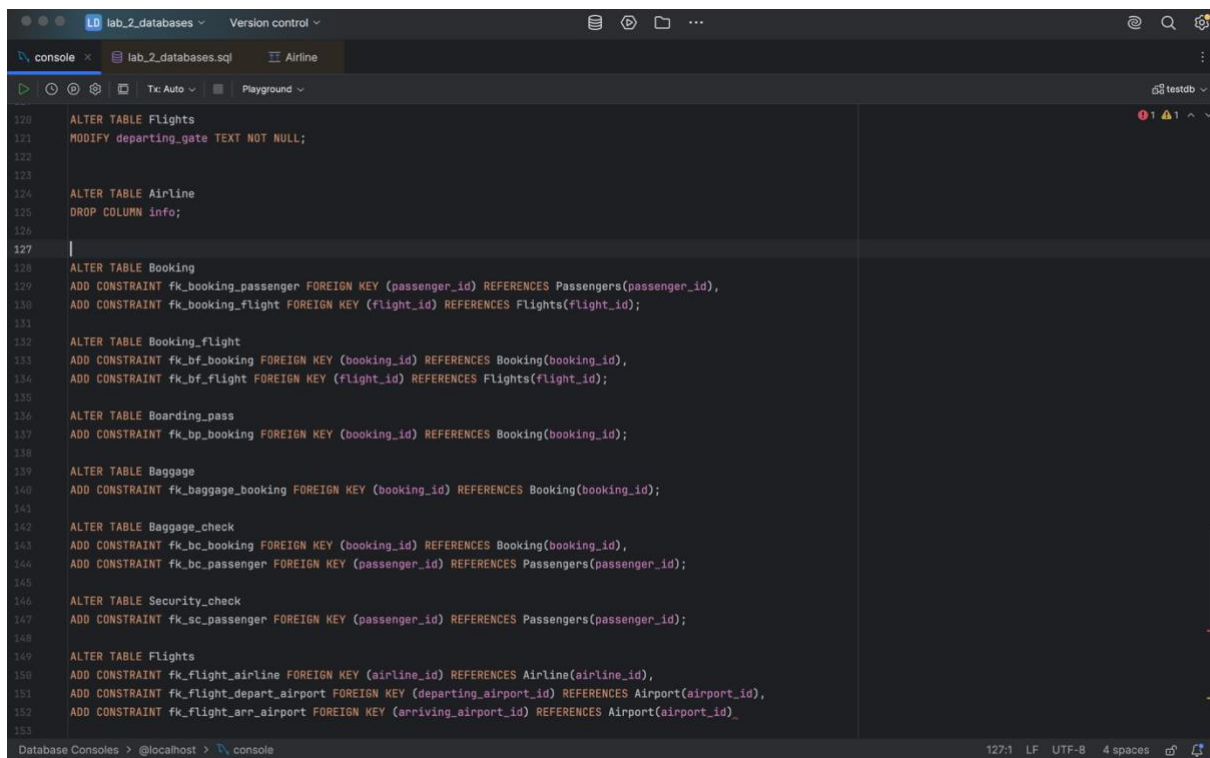
Drop the column info(varchar(50)) from the airline table.



```
102 ;
103
104
105 CREATE TABLE Security_check (
106     security_check_id INT PRIMARY KEY AUTO_INCREMENT,
107     check_result VARCHAR(20) NOT NULL,
108     created_at TIMESTAMP NOT NULL,
109     updated_at TIMESTAMP NOT NULL,
110     passenger_id INT NOT NULL
111 );
112
113 RENAME TABLE Airline_info TO Airline;
114
115
116 ALTER TABLE Booking
117 CHANGE price ticket_price DECIMAL(7,2) NOT NULL;
118
119
120 ALTER TABLE Flights
121 MODIFY departing_gate TEXT NOT NULL;
122
123
124 ALTER TABLE Airline
125 DROP COLUMN info;
126
127 |
128
129 ALTER TABLE Booking
130 ADD CONSTRAINT fk_booking_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id),
131 ADD CONSTRAINT fk_booking_flight FOREIGN KEY (flight_id) REFERENCES Flights(flight_id);
132
133 ALTER TABLE Booking_flight
134 ADD CONSTRAINT fk_bf_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
135 ADD CONSTRAINT fk_bf_flight FOREIGN KEY (flight_id) REFERENCES Flights(flight_id);
136
137 ALTER TABLE Booking_flight
```

Database Consoles > @localhost > console 127:1 LF UTF-8 4 spaces

Relationship between tables



```
lab_2_databases  Version control
console  lab_2_databases.sql  Airline

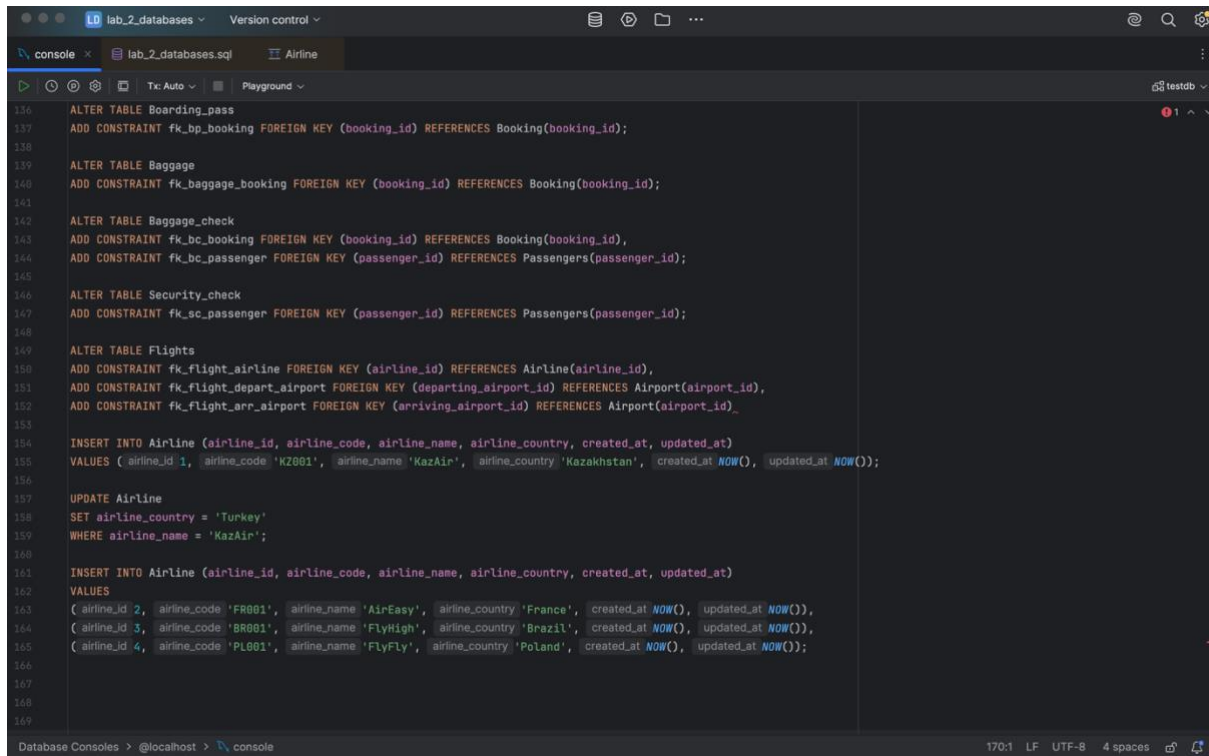
120 ALTER TABLE Flights
121 MODIFY departing_gate TEXT NOT NULL;
122
123
124 ALTER TABLE Airline
125 DROP COLUMN info;
126
127
128 ALTER TABLE Booking
129 ADD CONSTRAINT fk_booking_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id),
130 ADD CONSTRAINT fk_booking_flight FOREIGN KEY (flight_id) REFERENCES Flights(flight_id);
131
132 ALTER TABLE Booking_flight
133 ADD CONSTRAINT fk_bf_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
134 ADD CONSTRAINT fk_bf_flight FOREIGN KEY (flight_id) REFERENCES Flights(flight_id);
135
136 ALTER TABLE Boarding_pass
137 ADD CONSTRAINT fk_bp_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id);
138
139 ALTER TABLE Baggage
140 ADD CONSTRAINT fk_baggage_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id);
141
142 ALTER TABLE Baggage_check
143 ADD CONSTRAINT fk_bc_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
144 ADD CONSTRAINT fk_bc_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id);
145
146 ALTER TABLE Security_check
147 ADD CONSTRAINT fk_sc_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id);
148
149 ALTER TABLE Flights
150 ADD CONSTRAINT fk_flight_airline FOREIGN KEY (airline_id) REFERENCES Airline(airline_id),
151 ADD CONSTRAINT fk_flight_depart_airport FOREIGN KEY (departing_airport_id) REFERENCES Airport(airport_id),
152 ADD CONSTRAINT fk_flight_arr_airport FOREIGN KEY (arriving_airport_id) REFERENCES Airport(airport_id);
153

Database Consoles > @localhost > console 127:1 LF UTF-8 4 spaces
```

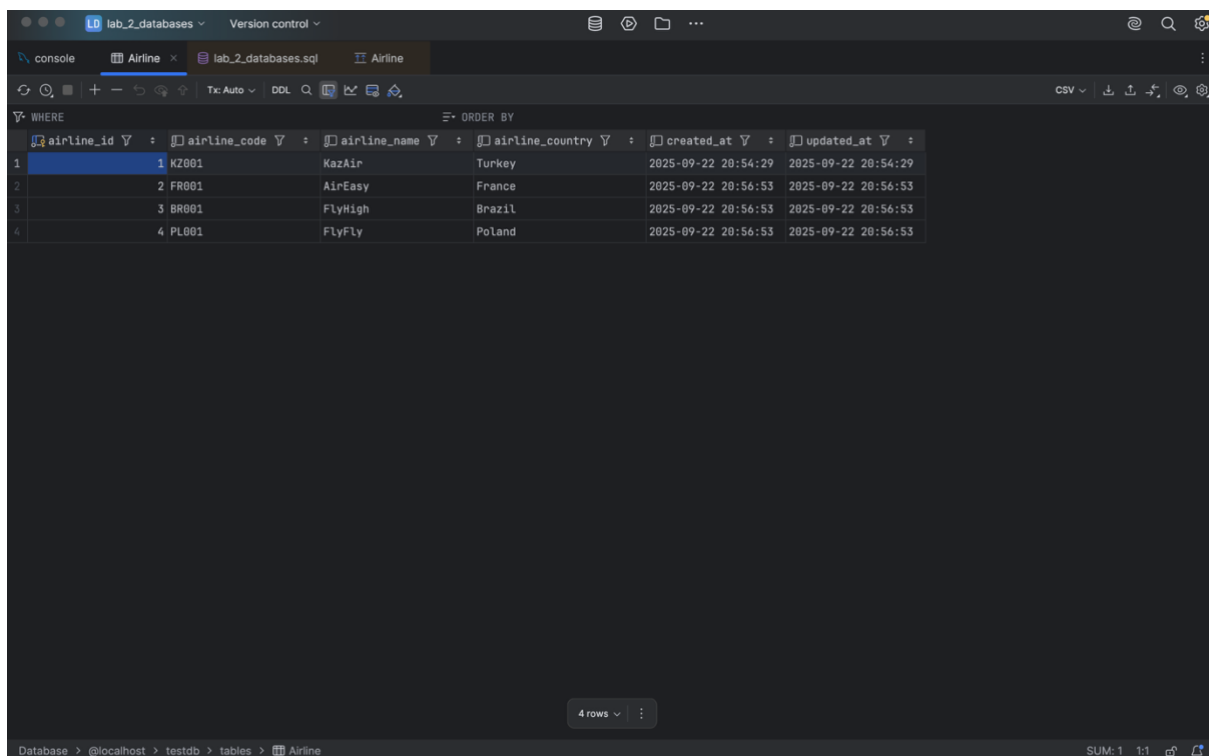
Add a new airline named "KazAir" based in "Kazakhstan" to the airline table.

Update the airline country "KazAir" to "Turkey".

Add three airlines at once: "AirEasy" in "France", "FlyHigh" in "Brazil" and "FlyFly" in "Poland".



```
136 ALTER TABLE Boarding_pass
137 ADD CONSTRAINT fk_bp_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id);
138
139 ALTER TABLE Baggage
140 ADD CONSTRAINT fk_baggage_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id);
141
142 ALTER TABLE Baggage_check
143 ADD CONSTRAINT fk_bc_booking FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
144 ADD CONSTRAINT fk_bc_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id);
145
146 ALTER TABLE Security_check
147 ADD CONSTRAINT fk_sc_passenger FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id);
148
149 ALTER TABLE Flights
150 ADD CONSTRAINT fk_flight_airline FOREIGN KEY (airline_id) REFERENCES Airline(airline_id),
151 ADD CONSTRAINT fk_flight_depart_airport FOREIGN KEY (departing_airport_id) REFERENCES Airport(airport_id),
152 ADD CONSTRAINT fk_flight_arr_airport FOREIGN KEY (arriving_airport_id) REFERENCES Airport(airport_id);
153
154 INSERT INTO Airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
155 VALUES (airline_id 1, airline_code 'KZ001', airline_name 'KazAir', airline_country 'Kazakhstan', created_at NOW(), updated_at NOW());
156
157 UPDATE Airline
158 SET airline_country = 'Turkey'
159 WHERE airline_name = 'KazAir';
160
161 INSERT INTO Airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
162 VALUES
163 (airline_id 2, airline_code 'FR001', airline_name 'AirEasy', airline_country 'France', created_at NOW(), updated_at NOW()),
164 (airline_id 3, airline_code 'BR001', airline_name 'FlyHigh', airline_country 'Brazil', created_at NOW(), updated_at NOW()),
165 (airline_id 4, airline_code 'PL001', airline_name 'FlyFly', airline_country 'Poland', created_at NOW(), updated_at NOW());
166
167
168
169
```



airline_id	airline_code	airline_name	airline_country	created_at	updated_at
1	KZ001	KazAir	Turkey	2025-09-22 20:54:29	2025-09-22 20:54:29
2	FR001	AirEasy	France	2025-09-22 20:56:53	2025-09-22 20:56:53
3	BR001	FlyHigh	Brazil	2025-09-22 20:56:53	2025-09-22 20:56:53
4	PL001	FlyFly	Poland	2025-09-22 20:56:53	2025-09-22 20:56:53

3. Conclusion

During this lab work, we learned how to create and edit tables and work with their data. DML operations demonstrated how to insert, update, and delete records, as well as how to safely apply bulk updates. The final ER schema provides a clear view of the system and supports future database operations.