

Laboratory work #2

```
1. CREATE TABLE airline_info (  
    airline_id SERIAL PRIMARY KEY,  
    airline_code VARCHAR(30) UNIQUE NOT NULL,  
    airline_name VARCHAR(50) NOT NULL,  
    airline_country VARCHAR(50) NOT NULL,  
    created_at TIMESTAMP DEFAULT NOW(),  
    updated_at TIMESTAMP DEFAULT NOW(),  
    info VARCHAR(50)  
);
```

```
CREATE TABLE Airport (  
    airport_id SERIAL PRIMARY KEY,  
    airport_name VARCHAR(50) NOT NULL,  
    country VARCHAR(50) NOT NULL,  
    state VARCHAR(50) NOT NULL,  
    city varchar(50) not null,  
    created_at timestamp default now(),  
    updated_at timestamp default now()  
);
```

```
create table Passengers (  
    passenger_id serial primary key,  
    first_name varchar(50) not null,  
    last_name varchar(50) not null,  
    date_of_birth date not null,  
    gender varchar(50) not null,  
    country_of_citizenship varchar(50) not null,  
    country_of_residence varchar(50) not null,  
    passport_number varchar(20) unique not null,  
    created_at timestamp default now(),  
    updated_at timestamp default now()  
);
```

```

create table Flights (
    flight_id serial primary key,
    sch_departure_time timestamp not null,
    sch_arrival_time timestamp not null,
    departing_airport_id int not null,
    arriving_airport_id int not null,
    departing_gate varchar(50) not null,
    arriving_gate varchar(50) not null,
    airline_id int not null,
    act_departure_time timestamp not null,
    act_arrival_time timestamp not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    constraint fk_departing_airport foreign key (departing_airport_id) references Airport(airport_id),
    constraint fk_arrival_airport foreign key (arriving_airport_id) references Airport(airport_id),
    constraint fk_airline foreign key (airline_id) references airline_info(airline_id)
);

```

```

create table Booking (
    booking_id serial primary key,
    flight_id int not null,
    passenger_id int not null,
    booking_platform varchar(50) not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    status varchar(50) not null,
    price decimal(7,2) not null,
    constraint fk_flight foreign key (flight_id) references Flights(flight_id),
    constraint fk_passenger foreign key (passenger_id) references Passengers(passenger_id)
);

```

```

create table Baggage (
    baggage_id serial primary key,
    weight_in_kg decimal(4, 2) not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    booking_id int not null,
    constraint fk_booking foreign key (booking_id) references Booking(booking_id)
);

```

```

create table Baggage_check (
    baggage_check_id serial primary key,
    check_result varchar(50) not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    booking_id int not null,
    passenger_id int not null,
    constraint fk_booking foreign key (booking_id) references Booking(booking_id),
    constraint fk_passenger foreign key (passenger_id) references Passengers(passenger_id)
);

```

```

create table Boarding_pass (
    boarding_pass_id serial primary key,
    booking_id int not null,
    seat varchar(50) not null,
    boarding_time timestamp not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    constraint fk_booking foreign key (booking_id) references Booking(booking_id)
);

```

```

create table Booking_flight (
    booking_flight_id serial primary key,
    booking_id int not null,
    flight_id int not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    constraint fk_booking foreign key (booking_id) references Booking(booking_id),
    constraint fk_flight foreign key (flight_id) references Flights(flight_id)
);

```

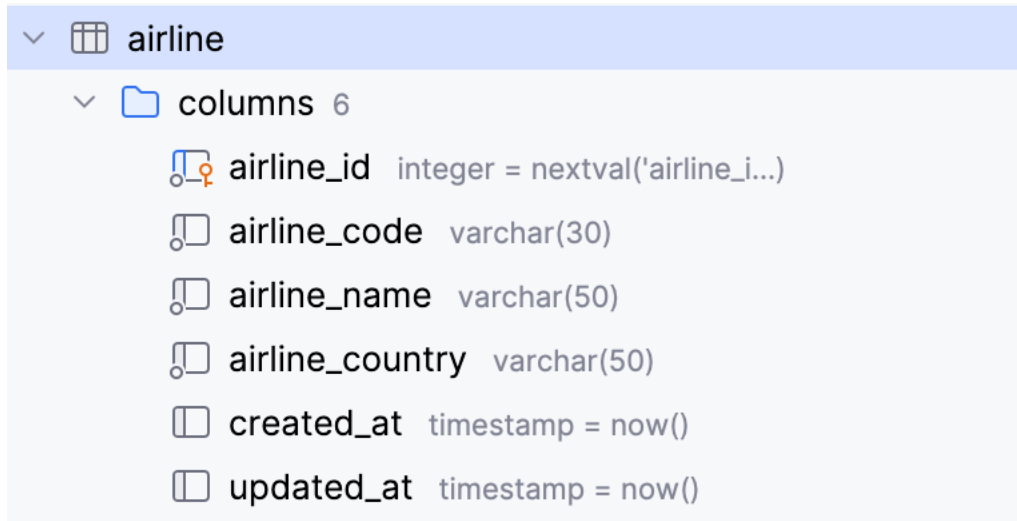
```

create table Security_check (
    security_check_id serial primary key,
    check_result varchar(20) not null,
    created_at timestamp default now(),
    updated_at timestamp default now(),
    passenger_id int not null,
    constraint fk_passenger foreign key (passenger_id) references Passengers(passenger_id)
);

```

2. All primary keys are defined (e.g. “serial primary key” in the Security_table).
3. For all attributes not null constraints are defined (e.g. check_result varchar(30) not null in the Security_check table).
- 4.

```
alter table airline_info rename to airline;
```



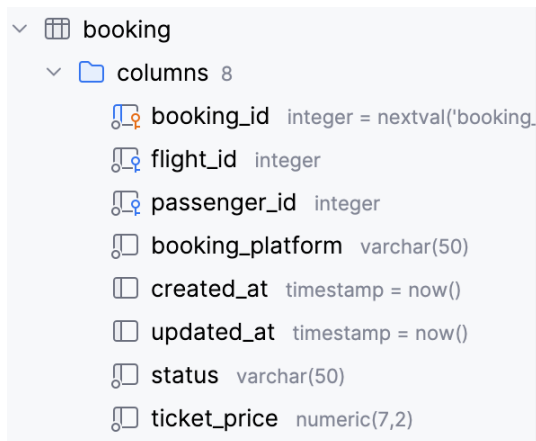
The screenshot shows a database schema viewer for a table named 'airline'. The table is expanded to show its columns. The columns are: 'airline_id' (integer, nextval('airline_i...)), 'airline_code' (varchar(30)), 'airline_name' (varchar(50)), 'airline_country' (varchar(50)), 'created_at' (timestamp = now()), and 'updated_at' (timestamp = now()).

Column Name	Data Type	Constraints
airline_id	integer	nextval('airline_i...')
airline_code	varchar(30)	
airline_name	varchar(50)	
airline_country	varchar(50)	
created_at	timestamp	now()
updated_at	timestamp	now()

Using ‘alter table’ as shown in the picture, the table ‘airline_info’ was renamed to ‘airline’.

- 5.

```
alter table booking  
rename column price to ticket_price;
```



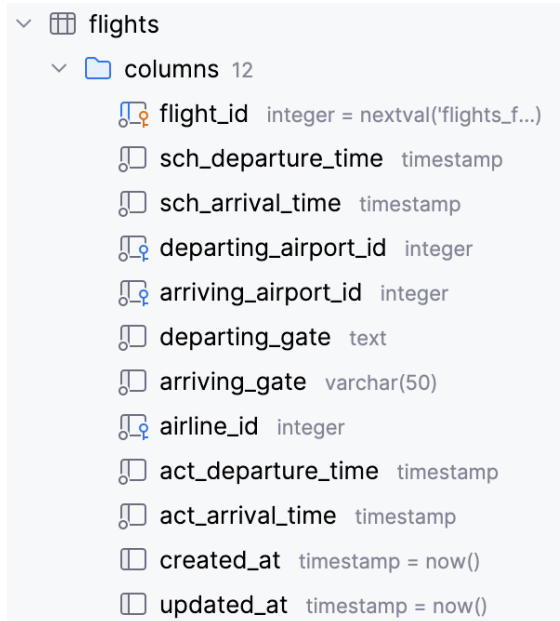
The screenshot shows a database schema viewer for a table named 'booking'. The table is expanded to show its columns. The columns are: 'booking_id' (integer, nextval('booking.')), 'flight_id' (integer), 'passenger_id' (integer), 'booking_platform' (varchar(50)), 'created_at' (timestamp = now()), 'updated_at' (timestamp = now()), 'status' (varchar(50)), and 'ticket_price' (numeric(7,2)).

Column Name	Data Type	Constraints
booking_id	integer	nextval('booking.'
flight_id	integer	
passenger_id	integer	
booking_platform	varchar(50)	
created_at	timestamp	now()
updated_at	timestamp	now()
status	varchar(50)	
ticket_price	numeric(7,2)	

Using ‘alter table’ and ‘rename’, the ‘price’ column’ was renamed to ‘ticket_price’.

6.

```
alter table Flights  
alter column departing_gate type text
```



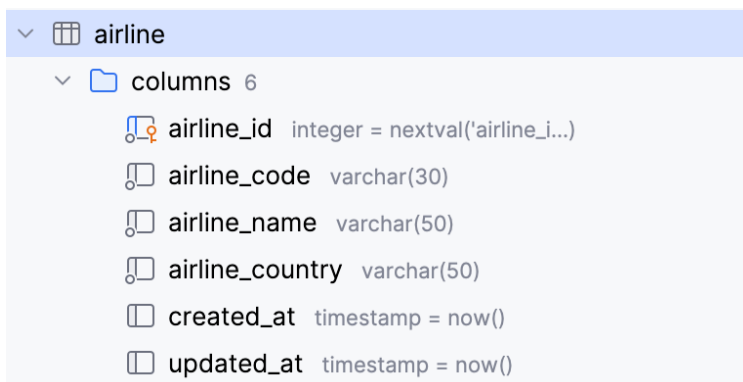
A screenshot of a database schema viewer showing the structure of the 'flights' table. The table has 12 columns. The 'departing_gate' column is highlighted, showing its data type as 'text'.

Column Name	Data Type
flight_id	integer = nextval('flights_f...')
sch_departure_time	timestamp
sch_arrival_time	timestamp
departing_airport_id	integer
arriving_airport_id	integer
departing_gate	text
arriving_gate	varchar(50)
airline_id	integer
act_departure_time	timestamp
act_arrival_time	timestamp
created_at	timestamp = now()
updated_at	timestamp = now()

Using 'alter table' and 'alter column', 'type' the 'departing_gate' column's data type was changed from 'varchar(50)' to 'text'.

7.

```
alter table airline  
drop column info;
```

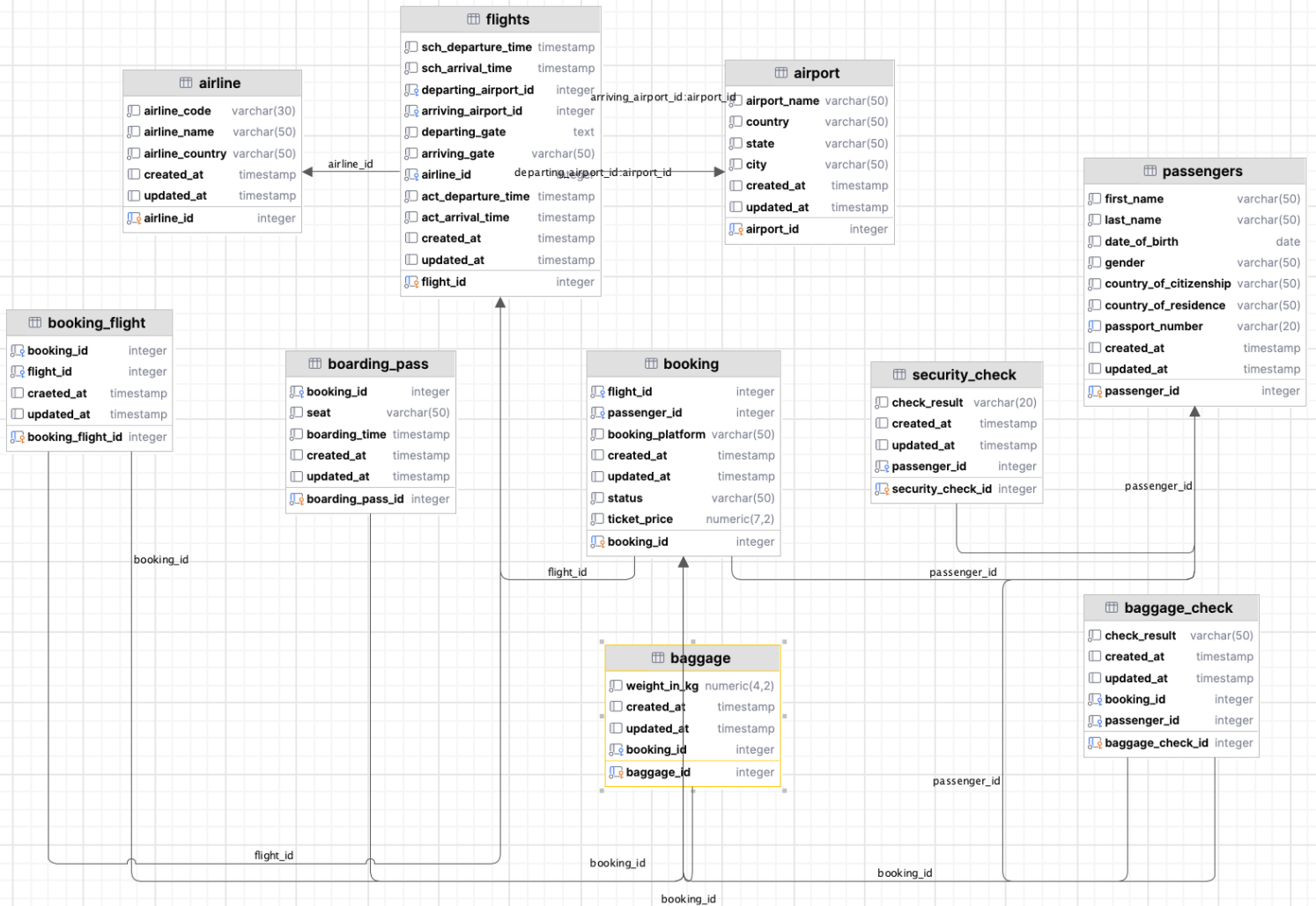


A screenshot of a database schema viewer showing the structure of the 'airline' table. The table has 6 columns. The 'info' column has been removed, and the 'created_at' and 'updated_at' columns are shown with their default values.

Column Name	Data Type
airline_id	integer = nextval('airline_i...')
airline_code	varchar(30)
airline_name	varchar(50)
airline_country	varchar(50)
created_at	timestamp = now()
updated_at	timestamp = now()

Using 'drop column' the 'info' column was deleted from the table 'airline'.

8.



Passengers -> Security_check, Booking, Baggage_check

- Passengers -> Security_check (1:M) passenger_id (one passenger can go through several check-ups)
- Passenger -> Booking (1:M) passenger_id (one passenger can have several bookings)
- Passenger -> Baggage_check (1:M) passenger_id (one passenger can have many baggage and baggage checks)

Booking -> Baggage_check, Baggage, Boarding_pass, Booking_flight

- Booking -> Baggage_check (1:M) booking_id (one booking can have many baggage and baggage check, respectfully)
- Booking -> Baggage (1:M) booking_id (one booking can have much baggage)
- Booking -> Boarding_pass (1:1) booking_id (one booking is related to exactly one boarding pass)
- Booking -> Booking_flight (1:M) booking_id (one booking can reserve several flights)

Flights -> Booking_flight

- Flights -> Booking_flight (1:M) flight_id (One flight can have many bookings)

Airport -> Flights

- Airport -> Flights (1:M) departing_airport_id (One airport can launch many flights, but airplanes can depart from one airport)
- Airport -> Flights (1:M) arriving_airport_id (One airport can receive many flights but one flight can land in only one airport)

Airline -> Flights

- Airline -> Flights (1:M) airline_id (one airline run several flights)

DML

1.

```
insert into airline (airline_code, airline_name, airline_country)
values ( airline_code 'KAZ', airline_name 'KazAIR', airline_country 'Kazakhstan');
```

‘insert into’ used to add data into the table.

2.

```
update airline
set airline_country = 'Turkey',
    updated_at = now()
where airline_name = 'KazAIR';
```

‘update’, and ‘set’ used to perform a change in a cell.

3.

```
insert into airline (airline_code, airline_name, airline_country)
values
( airline_code 'Af', airline_name 'AirEasy', airline_country 'France'),
( airline_code 'FB', airline_name 'FlyHigh', airline_country 'Brazil'),
( airline_code 'FP', airline_name 'FlyFly', airline_country 'Poland');
```

4.

```
delete from baggage_check
where booking_id in (
    select booking_id
    from booking
    where flight_id in (
        select flight_id from Flights
        where extract(year from act_arrival_time) = 2024
    )
);

delete from boarding_pass
where booking_id in (
    select booking_id
    from booking
    where flight_id in (
        select flight_id from Flights
        where extract(year from act_arrival_time) = 2024
    )
);

delete from baggage
where booking_id in (
    select booking_id
    from booking
    where flight_id in (
        select flight_id from flights
        where extract(year from act_arrival_time) = 2024
    )
);
```



```

delete from booking_flight
where booking_id in (
    select booking_id
    from booking
    where flight_id in (
        select flight_id from flights
        where extract(year from act_arrival_time) =2024
    )
);

delete from booking
where flight_id in (
    select flight_id from flights
    where extract(year from act_arrival_time) = 2024
);

delete from booking_flight
where flight_id in (
    select flight_id from flights
    where extract(year from act_arrival_time) = 2024
);

delete from flights
where extract(year from act_arrival_time) = 2024;

```

Tables are interconnected through foreign keys, thus specific rows should be deleted in a waterfall way.

5.

```

update booking
set ticket_price = ticket_price * 1.15
where true;

```

Used 'update', 'set' to set changes to the values.

6.

```
ALTER TABLE booking_flight
    DROP CONSTRAINT fk_booking,
    ADD CONSTRAINT fk_booking FOREIGN KEY (booking_id)
        REFERENCES booking(booking_id)
        ON DELETE CASCADE;

ALTER TABLE boarding_pass
    DROP CONSTRAINT fk_booking,
    ADD CONSTRAINT fk_booking FOREIGN KEY (booking_id)
        REFERENCES booking(booking_id)
        ON DELETE CASCADE;

ALTER TABLE baggage_check
    DROP CONSTRAINT fk_booking,
    ADD CONSTRAINT fk_booking FOREIGN KEY (booking_id)
        REFERENCES booking(booking_id)
        ON DELETE CASCADE;

ALTER TABLE baggage
    DROP CONSTRAINT fk_booking,
    ADD CONSTRAINT fk_booking
        FOREIGN KEY (booking_id)
            REFERENCES booking(booking_id)
            ON DELETE CASCADE;

DELETE FROM booking
WHERE ticket_price < 10000;
```