**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Чувашский государственный университет имени И.Н. Ульянова»**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

***Технология разработки ПО***

**Лабораторная работа №7**

**«Создание физической модели данных»**

Авиаперевозки

**Выполнил:**

студент группы ИВТ-41-20

Галкин Д.С.

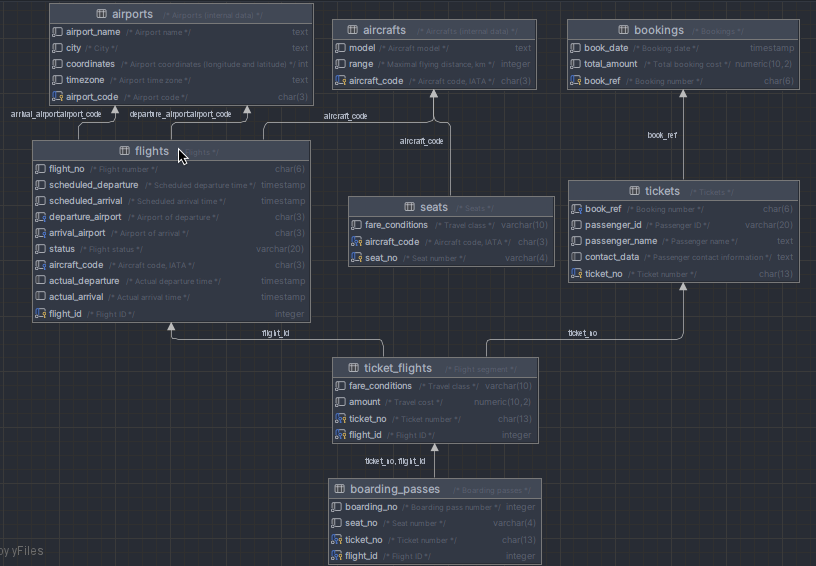
**Проверил:**

Ржавин В.В.

Чебоксары 2023

Цель работы: разработать физическую модель данных

**Диаграмма физической модели:**

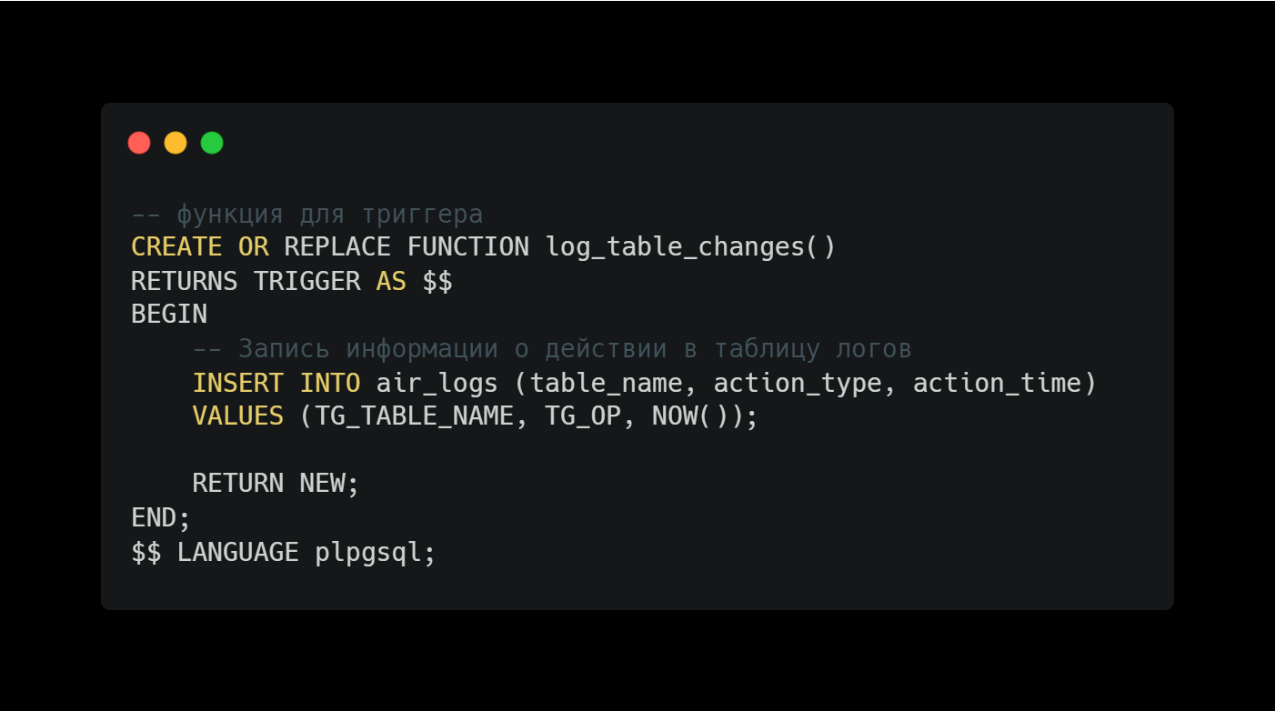


**Тексты разработанных триггеров и хранимых процедур:**

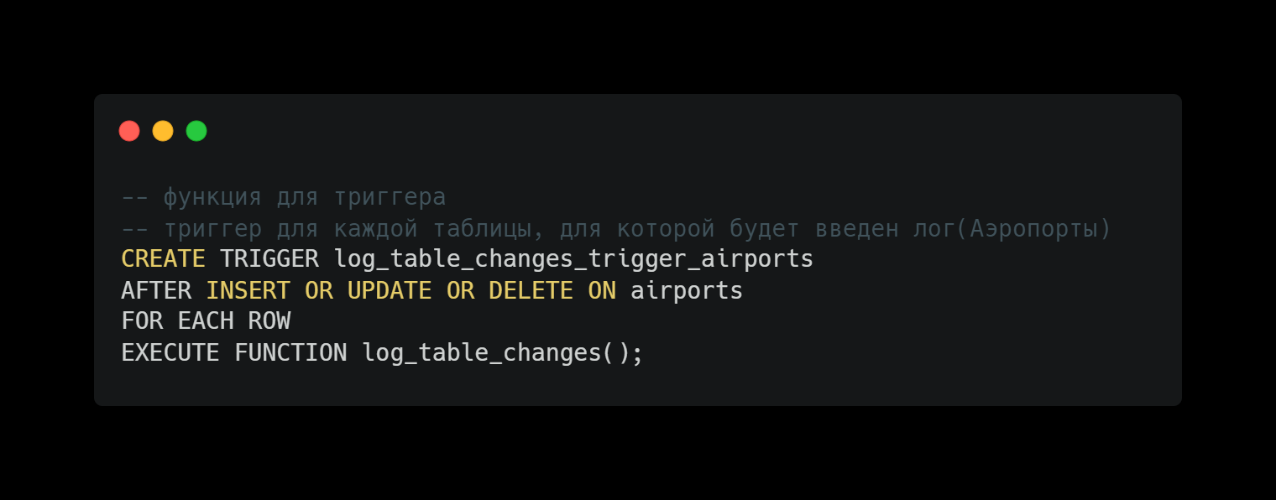
* Логирование
  + Создание дополнительной таблицы

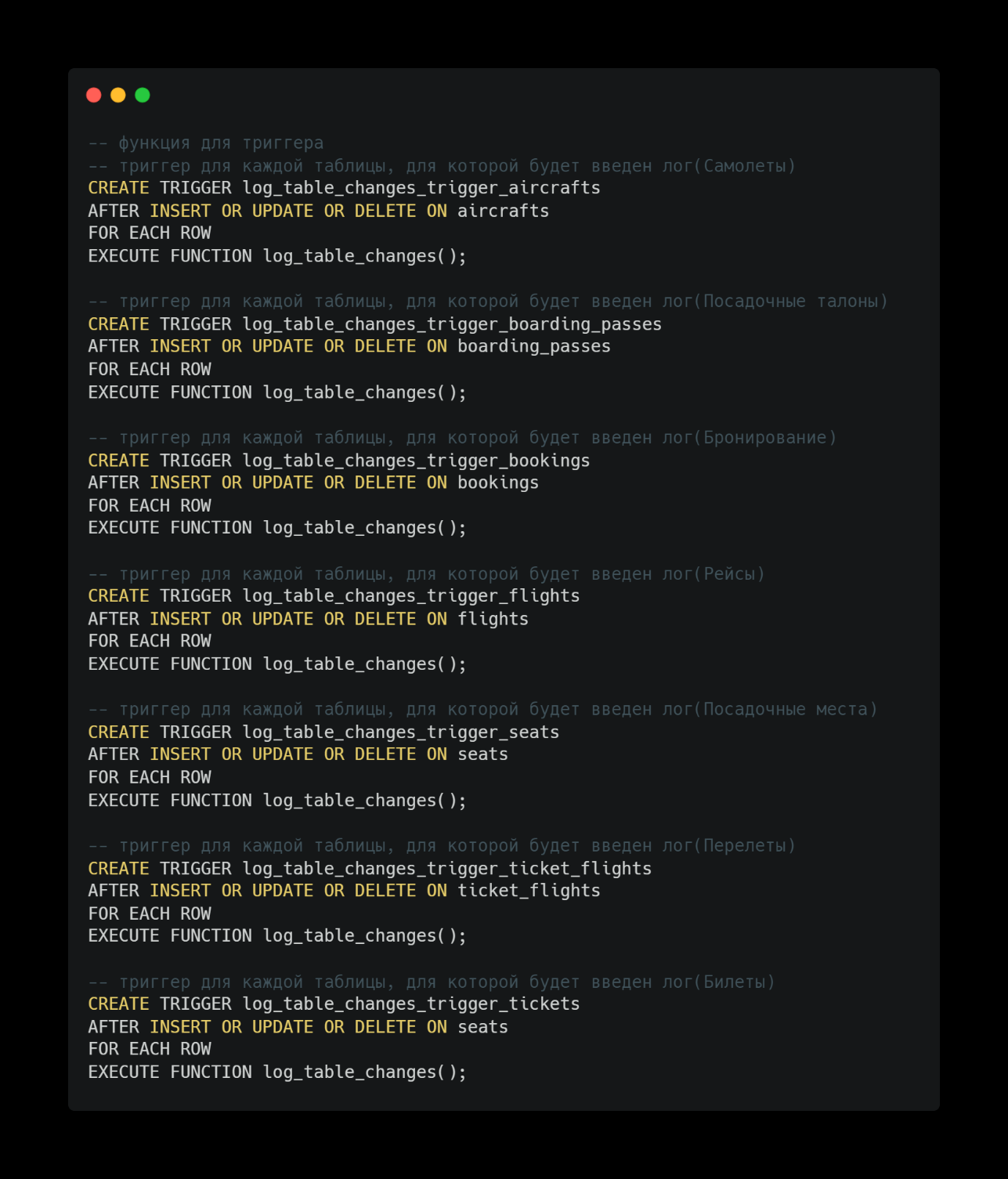


* + Реализация хранимой процедуры

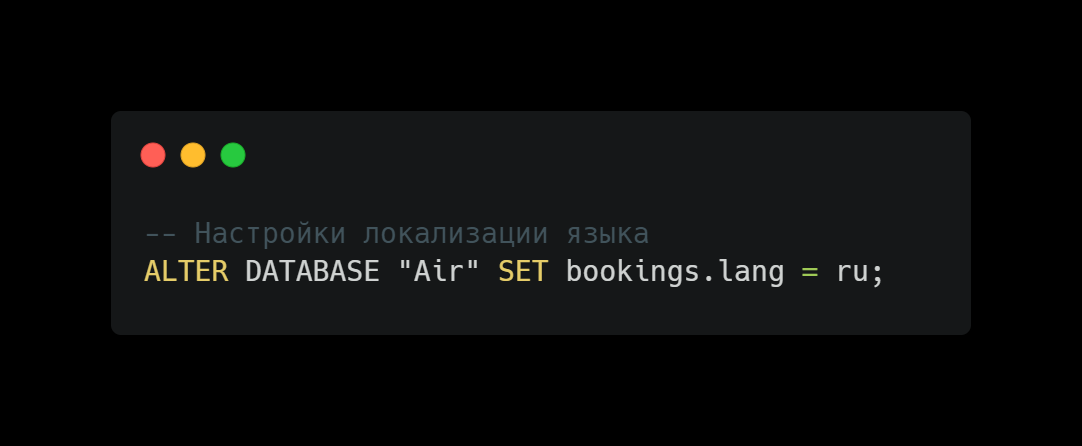


* + Триггеры логов для всех таблиц

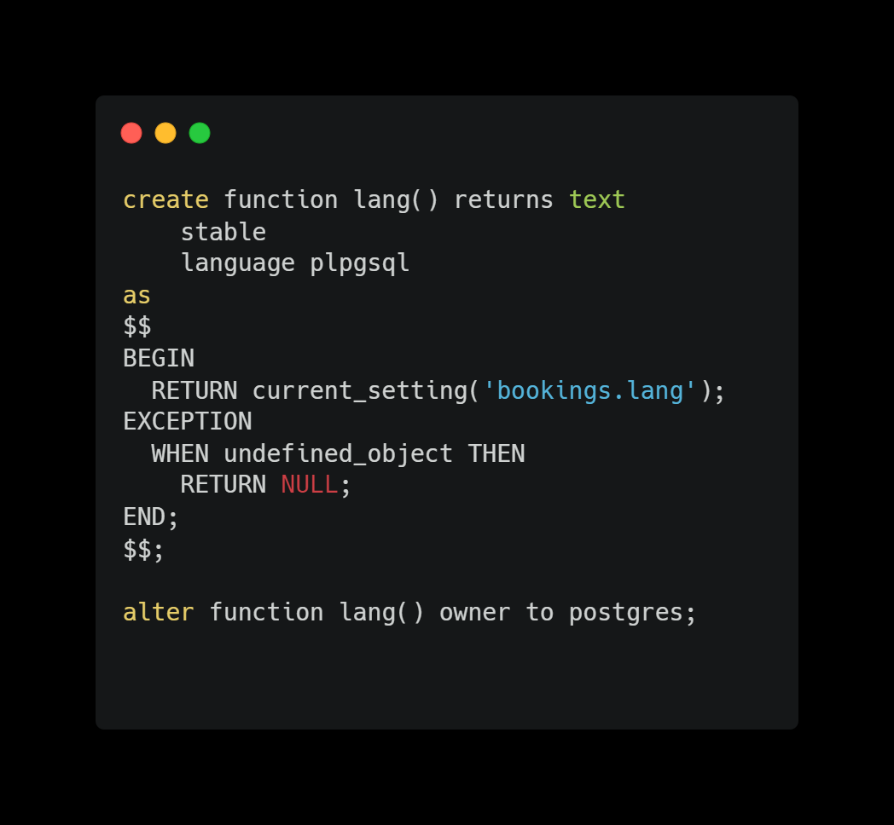




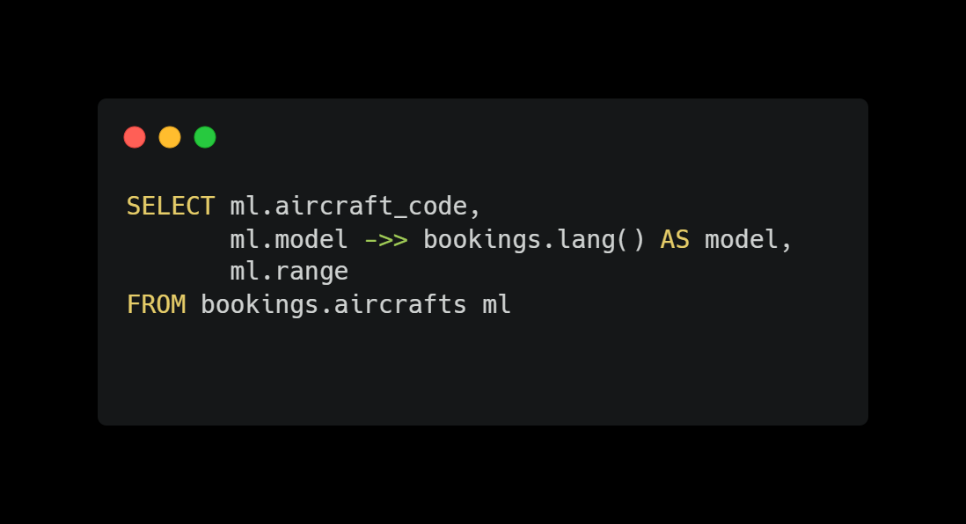
* Перевод типа Jsonb в текст
  + Настраиваем локализацию



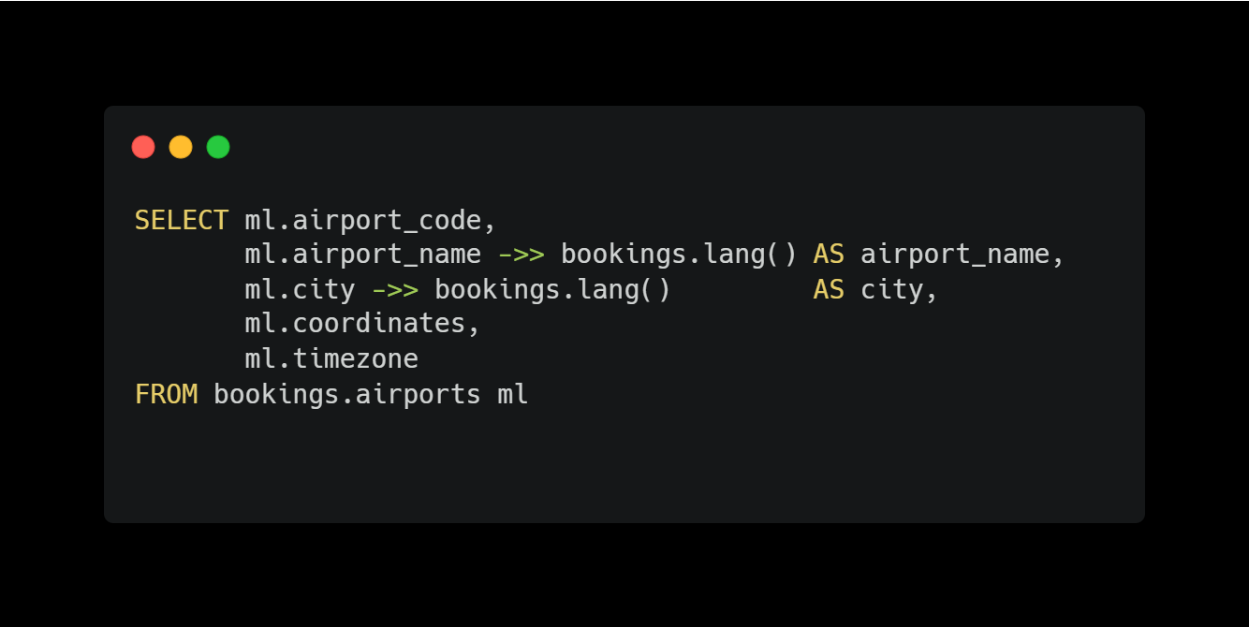
* + Создаем хранимую процедуру



* + Создаем View(aircrafts)



* + Создаем View(airports)



**PostgreSQL-код для создания базы данных в DataGrip:**

-- Database generated with pgModeler (PostgreSQL Database Modeler).  
-- pgModeler version: 1.1.0-alpha  
-- PostgreSQL version: 15.0  
-- Project Site: pgmodeler.io  
-- Model Author: ---  
-- -- object: pg\_database\_owner | type: ROLE --  
-- -- DROP ROLE IF EXISTS pg\_database\_owner;  
-- CREATE ROLE pg\_database\_owner WITH   
-- INHERIT  
-- PASSWORD '\*\*\*\*\*\*\*\*';  
-- -- ddl-end --  
--   
  
-- Database creation must be performed outside a multi lined SQL file.   
-- These commands were put in this file only as a convenience.  
--   
-- object: "Air\_test" | type: DATABASE --  
-- DROP DATABASE IF EXISTS "Air\_test";  
CREATE DATABASE "Air\_test"  
 ENCODING = 'UTF8'  
 LC\_COLLATE = 'English\_United States.1251'  
 LC\_CTYPE = 'English\_United States.1251'  
 TABLESPACE = pg\_default  
 OWNER = postgres;  
-- ddl-end --  
  
  
SET check\_function\_bodies = false;  
-- ddl-end --  
  
-- object: bookings | type: SCHEMA --  
-- DROP SCHEMA IF EXISTS bookings CASCADE;  
CREATE SCHEMA bookings;  
-- ddl-end --  
ALTER SCHEMA bookings OWNER TO postgres;  
-- ddl-end --  
COMMENT ON SCHEMA bookings IS E'Airlines demo database schema';  
-- ddl-end --  
  
-- Appended SQL commands --  
ALTER TABLE bookings  
ADD COLUMN test\_amount type\_amount;  
-- ddl-end --  
  
SET search\_path TO pg\_catalog,public,bookings;  
-- ddl-end --  
  
-- object: bookings.lang | type: FUNCTION --  
-- DROP FUNCTION IF EXISTS bookings.lang() CASCADE;  
CREATE FUNCTION bookings.lang ()  
 RETURNS text  
 LANGUAGE plpgsql  
 STABLE   
 CALLED ON NULL INPUT  
 SECURITY INVOKER  
 PARALLEL UNSAFE  
 COST 100  
 AS $$  
BEGIN  
 RETURN current\_setting('bookings.lang');  
EXCEPTION  
 WHEN undefined\_object THEN  
 RETURN NULL;  
END;  
$$;  
-- ddl-end --  
ALTER FUNCTION bookings.lang() OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.now | type: FUNCTION --  
-- DROP FUNCTION IF EXISTS bookings.now() CASCADE;  
CREATE FUNCTION bookings.now ()  
 RETURNS timestamp with time zone  
 LANGUAGE sql  
 IMMUTABLE   
 CALLED ON NULL INPUT  
 SECURITY INVOKER  
 PARALLEL UNSAFE  
 COST 100  
 AS $$  
SELECT '2017-08-15 18:00:00'::TIMESTAMP AT TIME ZONE 'Europe/Moscow';  
$$;  
-- ddl-end --  
ALTER FUNCTION bookings.now() OWNER TO postgres;  
-- ddl-end --  
COMMENT ON FUNCTION bookings.now() IS E'Point in time according to which the data are generated';  
-- ddl-end --  
  
-- object: bookings.aircrafts | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.aircrafts CASCADE;  
CREATE TABLE bookings.aircrafts (  
 aircraft\_code character(3) NOT NULL,  
 model text NOT NULL,  
 range integer NOT NULL,  
 CONSTRAINT aircrafts\_range\_check CHECK ((range > 0)),  
 CONSTRAINT aircrafts\_pkey PRIMARY KEY (aircraft\_code)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.aircrafts IS E'Aircrafts (internal data)';  
-- ddl-end --  
COMMENT ON COLUMN bookings.aircrafts.aircraft\_code IS E'Aircraft code, IATA';  
-- ddl-end --  
COMMENT ON COLUMN bookings.aircrafts.model IS E'Aircraft model';  
-- ddl-end --  
COMMENT ON COLUMN bookings.aircrafts.range IS E'Maximal flying distance, km';  
-- ddl-end --  
ALTER TABLE bookings.aircrafts OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.airports | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.airports CASCADE;  
CREATE TABLE bookings.airports (  
 airport\_code character(3) NOT NULL,  
 airport\_name text NOT NULL,  
 city text NOT NULL,  
 coordinates point NOT NULL,  
 timezone text NOT NULL,  
 CONSTRAINT airports\_data\_pkey PRIMARY KEY (airport\_code)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.airports IS E'Airports (internal data)';  
-- ddl-end --  
COMMENT ON COLUMN bookings.airports.airport\_code IS E'Airport code';  
-- ddl-end --  
COMMENT ON COLUMN bookings.airports.airport\_name IS E'Airport name';  
-- ddl-end --  
COMMENT ON COLUMN bookings.airports.city IS E'City';  
-- ddl-end --  
COMMENT ON COLUMN bookings.airports.coordinates IS E'Airport coordinates (longitude and latitude)';  
-- ddl-end --  
COMMENT ON COLUMN bookings.airports.timezone IS E'Airport time zone';  
-- ddl-end --  
ALTER TABLE bookings.airports OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.boarding\_passes | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.boarding\_passes CASCADE;  
CREATE TABLE bookings.boarding\_passes (  
 ticket\_no character(13) NOT NULL,  
 flight\_id integer NOT NULL,  
 boarding\_no integer NOT NULL,  
 seat\_no character varying(4) NOT NULL,  
 CONSTRAINT boarding\_passes\_pkey PRIMARY KEY (ticket\_no,flight\_id)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.boarding\_passes IS E'Boarding passes';  
-- ddl-end --  
COMMENT ON COLUMN bookings.boarding\_passes.ticket\_no IS E'Ticket number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.boarding\_passes.flight\_id IS E'Flight ID';  
-- ddl-end --  
COMMENT ON COLUMN bookings.boarding\_passes.boarding\_no IS E'Boarding pass number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.boarding\_passes.seat\_no IS E'Seat number';  
-- ddl-end --  
ALTER TABLE bookings.boarding\_passes OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.bookings | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.bookings CASCADE;  
CREATE TABLE bookings.bookings (  
 book\_ref character(6) NOT NULL,  
 book\_date timestamp NOT NULL,  
 total\_amount numeric(10,2) NOT NULL,  
 CONSTRAINT bookings\_pkey PRIMARY KEY (book\_ref)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.bookings IS E'Bookings';  
-- ddl-end --  
COMMENT ON COLUMN bookings.bookings.book\_ref IS E'Booking number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.bookings.book\_date IS E'Booking date';  
-- ddl-end --  
COMMENT ON COLUMN bookings.bookings.total\_amount IS E'Total booking cost';  
-- ddl-end --  
ALTER TABLE bookings.bookings OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.flights\_flight\_id\_seq | type: SEQUENCE --  
-- DROP SEQUENCE IF EXISTS bookings.flights\_flight\_id\_seq CASCADE;  
CREATE SEQUENCE bookings.flights\_flight\_id\_seq  
 INCREMENT BY 1  
 MINVALUE 1  
 MAXVALUE 9223372036854775807  
 START WITH 1  
 CACHE 1  
 NO CYCLE  
 OWNED BY NONE;  
  
-- ddl-end --  
ALTER SEQUENCE bookings.flights\_flight\_id\_seq OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.flights | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.flights CASCADE;  
CREATE TABLE bookings.flights (  
 flight\_id integer NOT NULL DEFAULT nextval('flights\_flight\_id\_seq'::regclass),  
 flight\_no character(6) NOT NULL,  
 scheduled\_departure timestamp NOT NULL,  
 scheduled\_arrival timestamp NOT NULL,  
 departure\_airport character(3) NOT NULL,  
 arrival\_airport character(3) NOT NULL,  
 status character varying(20) NOT NULL,  
 aircraft\_code character(3) NOT NULL,  
 actual\_departure timestamp,  
 actual\_arrival timestamp,  
 CONSTRAINT flights\_pkey PRIMARY KEY (flight\_id)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.flights IS E'Flights';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.flight\_id IS E'Flight ID';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.flight\_no IS E'Flight number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.scheduled\_departure IS E'Scheduled departure time';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.scheduled\_arrival IS E'Scheduled arrival time';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.departure\_airport IS E'Airport of departure';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.arrival\_airport IS E'Airport of arrival';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.status IS E'Flight status';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.aircraft\_code IS E'Aircraft code, IATA';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.actual\_departure IS E'Actual departure time';  
-- ddl-end --  
COMMENT ON COLUMN bookings.flights.actual\_arrival IS E'Actual arrival time';  
-- ddl-end --  
ALTER TABLE bookings.flights OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.seats | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.seats CASCADE;  
CREATE TABLE bookings.seats (  
 aircraft\_code character(3) NOT NULL,  
 seat\_no character varying(4) NOT NULL,  
 fare\_conditions character varying(10) NOT NULL,  
 CONSTRAINT seats\_fare\_conditions\_check CHECK (((fare\_conditions)::text = ANY (ARRAY[('Economy'::character varying)::text, ('Comfort'::character varying)::text, ('Business'::character varying)::text]))),  
 CONSTRAINT seats\_pkey PRIMARY KEY (aircraft\_code,seat\_no)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.seats IS E'Seats';  
-- ddl-end --  
COMMENT ON COLUMN bookings.seats.aircraft\_code IS E'Aircraft code, IATA';  
-- ddl-end --  
COMMENT ON COLUMN bookings.seats.seat\_no IS E'Seat number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.seats.fare\_conditions IS E'Travel class';  
-- ddl-end --  
ALTER TABLE bookings.seats OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.ticket\_flights | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.ticket\_flights CASCADE;  
CREATE TABLE bookings.ticket\_flights (  
 ticket\_no character(13) NOT NULL,  
 flight\_id integer NOT NULL,  
 fare\_conditions character varying(10) NOT NULL,  
 amount numeric(10,2) NOT NULL,  
 CONSTRAINT ticket\_flights\_pkey PRIMARY KEY (ticket\_no,flight\_id)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.ticket\_flights IS E'Flight segment';  
-- ddl-end --  
COMMENT ON COLUMN bookings.ticket\_flights.ticket\_no IS E'Ticket number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.ticket\_flights.flight\_id IS E'Flight ID';  
-- ddl-end --  
COMMENT ON COLUMN bookings.ticket\_flights.fare\_conditions IS E'Travel class';  
-- ddl-end --  
COMMENT ON COLUMN bookings.ticket\_flights.amount IS E'Travel cost';  
-- ddl-end --  
ALTER TABLE bookings.ticket\_flights OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.tickets | type: TABLE --  
-- DROP TABLE IF EXISTS bookings.tickets CASCADE;  
CREATE TABLE bookings.tickets (  
 ticket\_no character(13) NOT NULL,  
 book\_ref character(6) NOT NULL,  
 passenger\_id character varying(20) NOT NULL,  
 passenger\_name text NOT NULL,  
 contact\_data text,  
 CONSTRAINT tickets\_pkey PRIMARY KEY (ticket\_no)  
);  
-- ddl-end --  
COMMENT ON TABLE bookings.tickets IS E'Tickets';  
-- ddl-end --  
COMMENT ON COLUMN bookings.tickets.ticket\_no IS E'Ticket number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.tickets.book\_ref IS E'Booking number';  
-- ddl-end --  
COMMENT ON COLUMN bookings.tickets.passenger\_id IS E'Passenger ID';  
-- ddl-end --  
COMMENT ON COLUMN bookings.tickets.passenger\_name IS E'Passenger name';  
-- ddl-end --  
COMMENT ON COLUMN bookings.tickets.contact\_data IS E'Passenger contact information';  
-- ddl-end --  
ALTER TABLE bookings.tickets OWNER TO postgres;  
-- ddl-end --  
  
-- object: bookings.type\_amount | type: DOMAIN --  
-- DROP DOMAIN IF EXISTS bookings.type\_amount CASCADE;  
CREATE DOMAIN bookings.type\_amount AS numeric(10,2);  
-- ddl-end --  
ALTER DOMAIN bookings.type\_amount OWNER TO postgres;  
-- ddl-end --  
  
-- object: boarding\_passes\_ticket\_no\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.boarding\_passes DROP CONSTRAINT IF EXISTS boarding\_passes\_ticket\_no\_fkey CASCADE;  
ALTER TABLE bookings.boarding\_passes ADD CONSTRAINT boarding\_passes\_ticket\_no\_fkey FOREIGN KEY (ticket\_no,flight\_id)  
REFERENCES bookings.ticket\_flights (ticket\_no,flight\_id) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: flights\_aircraft\_code\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.flights DROP CONSTRAINT IF EXISTS flights\_aircraft\_code\_fkey CASCADE;  
ALTER TABLE bookings.flights ADD CONSTRAINT flights\_aircraft\_code\_fkey FOREIGN KEY (aircraft\_code)  
REFERENCES bookings.aircrafts (aircraft\_code) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: flights\_arrival\_airport\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.flights DROP CONSTRAINT IF EXISTS flights\_arrival\_airport\_fkey CASCADE;  
ALTER TABLE bookings.flights ADD CONSTRAINT flights\_arrival\_airport\_fkey FOREIGN KEY (arrival\_airport)  
REFERENCES bookings.airports (airport\_code) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: flights\_departure\_airport\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.flights DROP CONSTRAINT IF EXISTS flights\_departure\_airport\_fkey CASCADE;  
ALTER TABLE bookings.flights ADD CONSTRAINT flights\_departure\_airport\_fkey FOREIGN KEY (departure\_airport)  
REFERENCES bookings.airports (airport\_code) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: seats\_aircraft\_code\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.seats DROP CONSTRAINT IF EXISTS seats\_aircraft\_code\_fkey CASCADE;  
ALTER TABLE bookings.seats ADD CONSTRAINT seats\_aircraft\_code\_fkey FOREIGN KEY (aircraft\_code)  
REFERENCES bookings.aircrafts (aircraft\_code) MATCH SIMPLE  
ON DELETE CASCADE ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: ticket\_flights\_flight\_id\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.ticket\_flights DROP CONSTRAINT IF EXISTS ticket\_flights\_flight\_id\_fkey CASCADE;  
ALTER TABLE bookings.ticket\_flights ADD CONSTRAINT ticket\_flights\_flight\_id\_fkey FOREIGN KEY (flight\_id)  
REFERENCES bookings.flights (flight\_id) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: ticket\_flights\_ticket\_no\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.ticket\_flights DROP CONSTRAINT IF EXISTS ticket\_flights\_ticket\_no\_fkey CASCADE;  
ALTER TABLE bookings.ticket\_flights ADD CONSTRAINT ticket\_flights\_ticket\_no\_fkey FOREIGN KEY (ticket\_no)  
REFERENCES bookings.tickets (ticket\_no) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: tickets\_book\_ref\_fkey | type: CONSTRAINT --  
-- ALTER TABLE bookings.tickets DROP CONSTRAINT IF EXISTS tickets\_book\_ref\_fkey CASCADE;  
ALTER TABLE bookings.tickets ADD CONSTRAINT tickets\_book\_ref\_fkey FOREIGN KEY (book\_ref)  
REFERENCES bookings.bookings (book\_ref) MATCH SIMPLE  
ON DELETE NO ACTION ON UPDATE NO ACTION;  
-- ddl-end --  
  
-- object: "grant\_CU\_26541e8cda" | type: PERMISSION --  
GRANT CREATE,USAGE  
 ON SCHEMA public  
 TO pg\_database\_owner;  
-- ddl-end --  
  
-- object: "grant\_U\_cd8e46e7b6" | type: PERMISSION --  
GRANT USAGE  
 ON SCHEMA public  
 TO PUBLIC;  
-- ddl-end --

**Вывод**: в ходе выполнения данной лабораторной работы я разработал логическую модель данных «Авиоперевозки» в PGModeler.