

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2
«АНАЛИЗ ДАННЫХ.
ПОСТРОЕНИЕ ИНФОЛОГИЧЕСКОЙ МОДЕЛИ ДАННЫХ БД»
по дисциплине «Проектирование и реализация баз данных»**

Обучающиеся Меньшенин Евгений, Мищенко Максим
Факультет прикладной информатики
Группа К3241, К3239
Направление подготовки 09.03.03 Прикладная информатика
Образовательная программа Мобильные и сетевые технологии 2023
Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2024/2025

Цель работы: овладеть практическими навыками проведения анализа данных системы и построения инфологической модели данных БД методом «сущность-связь».

Практическое задание.

1. Проанализировать предметную область согласно варианту задания.
2. Выполнить инфологическое моделирование базы данных по заданной предметной области с использованием метода ER-диаграмм («сущность-связь») в комбинированной нотации Питера Чена - Кириллова (задание 1.1 варианта).
3. Реализовать разработанную ИЛМ в нотации IDEF1X.

Вариант 8. БД «Аэропорт» (допустимо к выполнению для команды из 2-х студентов)

Описание предметной области: Необходимо обеспечить продажу билетов на нужный рейс, при отсутствии билетов (необходимого количества билетов) предложить билет на ближайший рейс.

Рейсы выполняются по расписанию. Но есть рейсы назначаемые на определенный период или разовые.

Рейс может иметь несколько транзитных посадок (до 3-х).

На каждый рейс формируется экипаж из сотрудников компании, выполняющей рейс. В состав экипажа входят первый и второй пилоты, крью (старший стюард) и стюарды. Необходимо хранить данные о прохождении медосмотра перед рейсом (дата, статус, причина недопуска).

Билет может быть приобретен в кассе или онлайн. К базовой стоимости билета может быть дополнительная плата за выбор места, страховку багажа и т.п. Если билет приобретен в кассе, необходимо знать, в какой. Для каждой кассы известны номер и адрес. Кассы могут располагаться в различных населенных пунктах.

При покупке билета номер места может быть неизвестен пассажиру до регистрации на рейс.

БД должна содержать следующий минимальный набор сведений: Бортовой номер самолета. Тип самолета. Количество мест. Страна. Производитель. Грузоподъемность. Скорость. Дата выпуска. Налёт в часах. Дата последнего ремонта. Назначение самолета. Расход топлива. Код экипажа. Паспортные данные членов экипажа. Номер рейса. Дата вылета. Время вылета. Аэропорт вылета. Аэропорт назначения. Расстояние. Транзитные посадки (прилет, вылет, аэропорт, время в аэропорту). ФИО пассажира. Паспортные данные. Номер места. Тип места. Цена билета. Касса продажи билета (возможен электронный билет) (номер и адрес).

Дополните состав атрибутов на основе анализа предметной области.

Задание 1. Выполните инфологическое моделирование базы данных системы. (Ограничения задать самостоятельно.)

Задание 2. Создайте логическую модель БД, используя ИЛМ (задание 1.1). Используйте необходимые средства поддержки целостности данных в СУБД.

Порядок выполнения задания

1. Изучить предметную область
2. Выполнить следующие этапы моделирования структур данных:
 - 2.1. проанализировать состав объектов предметной области и атрибутов, их характеризующих;
 - 2.2. определить вычисляемые атрибуты, которые можно не хранить в БД;
 - 2.3. выделить сущности:
 - 2.3.1. стержневые;
 - 2.3.2. зависимые: выделяются на основе анализа многозначных или составных свойств сущности;
 - 2.3.3. обозначающие;
 - 2.4. выделить связи: ассоциации между сущностями.

3. Проанализировать свойства связей:
 - 3.1. определить тип связи и классы принадлежности сущностей (максимальную и минимальную кардинальность связей);
 - 3.2. проанализировать свойства связей для характеристических и обозначающих сущностей: определить тип связи и классы принадлежности сущностей;
 - 3.3. проанализировать свойства связей: выделить атрибуты связей и определить ключи (первичные и внешние);
 - 3.4. представить состав реквизитов сущностей в виде “название сущности (перечень реквизитов)”. Например: Студент (ID студента, фамилия, имя, отчество, номер группы, дата рождения, стипендия, оценки).
4. На основе проведенного анализа построить схему инфологической модели в виде схемы данных (диаграммы ER-типов) в рассмотренной нотации Питера Чена.
5. Выполнить моделирование в среде CA ERwin Data Modeler (создать модель Logical/Physical).
6. Дать характеристику атрибутов сущностей (типизация (структурная часть) и их ограничения (целостная часть)) по следующей схеме (словарь данных). Для внешних ключей необходимо указать правила соответствия первичным ключам (для связываемых сущностей). Дать характеристику в виде таблицы согласно свойствам элементов модели в CA ERwin Data Modeler.
7. Описать алгоритмические связи показателей (вычисляемые атрибуты) при наличии.

Ход работы

Название создаваемой БД - “Аэропорт”.

Ниже описан **состав реквизитов сущностей**.

1. CREW (**crew_id**, *member_id*, role, medical_check_datetime, medical_status, medical_reason, *flight_id*)
2. CREW_MEMBERS (**member_id**, *company_id*, full_name, passport_serial, passport_number, passport_region, role)
3. COMPANY (**company_id**, name, country)
4. PLANES (**plane_id**, *model_id*, status, flight_hours, *company_id*, last_maintenance_datetime)
5. PLANE_MODELS (**model_id**, title, engines, fuel_consumption, speed, flight_range, cargo_capacity, seats, seat_count)
6. FLIGHTS (**flight_id**, *plane_id*, *route_id*, *crew_id*, status, *departure_airport*, *destination_airport*, distance, departure_datetime_real, arrival_datetime_real)
7. SCHEDULE (**route_id**, *departure_airport*, *destination_airport*, departure_datetime, arrival_datetime)
8. AIRPORTS (**airport_code**, country, status, city, name)
9. TICKETS (**ticket_id**, *flight_id*, *passenger_id*, *seat_id*, sale_channel, *cash_register_id*, additional_fee, status)
10. PASSENGERS (**passenger_id**, full_name, passport_serial, passport_number, passport_region, birth_date)
11. SEATS (**seat_id**, *flight_id*, seat_number, seat_type, base_price, status)
12. CASH_REGISTERS (**cash_register_id**, address, status)
13. TRANST_FACT (**transit_fact_id**, *transit_id*, *flight_id*, departure_datetime_fact)
14. TRANSIT_STOPS (**transit_id**, *airport_id*, arrival_datetime, departure_datetime)

На рис. 1 изображена схема инфологической модели данных БД в нотации Питера Чена-Кириллова.

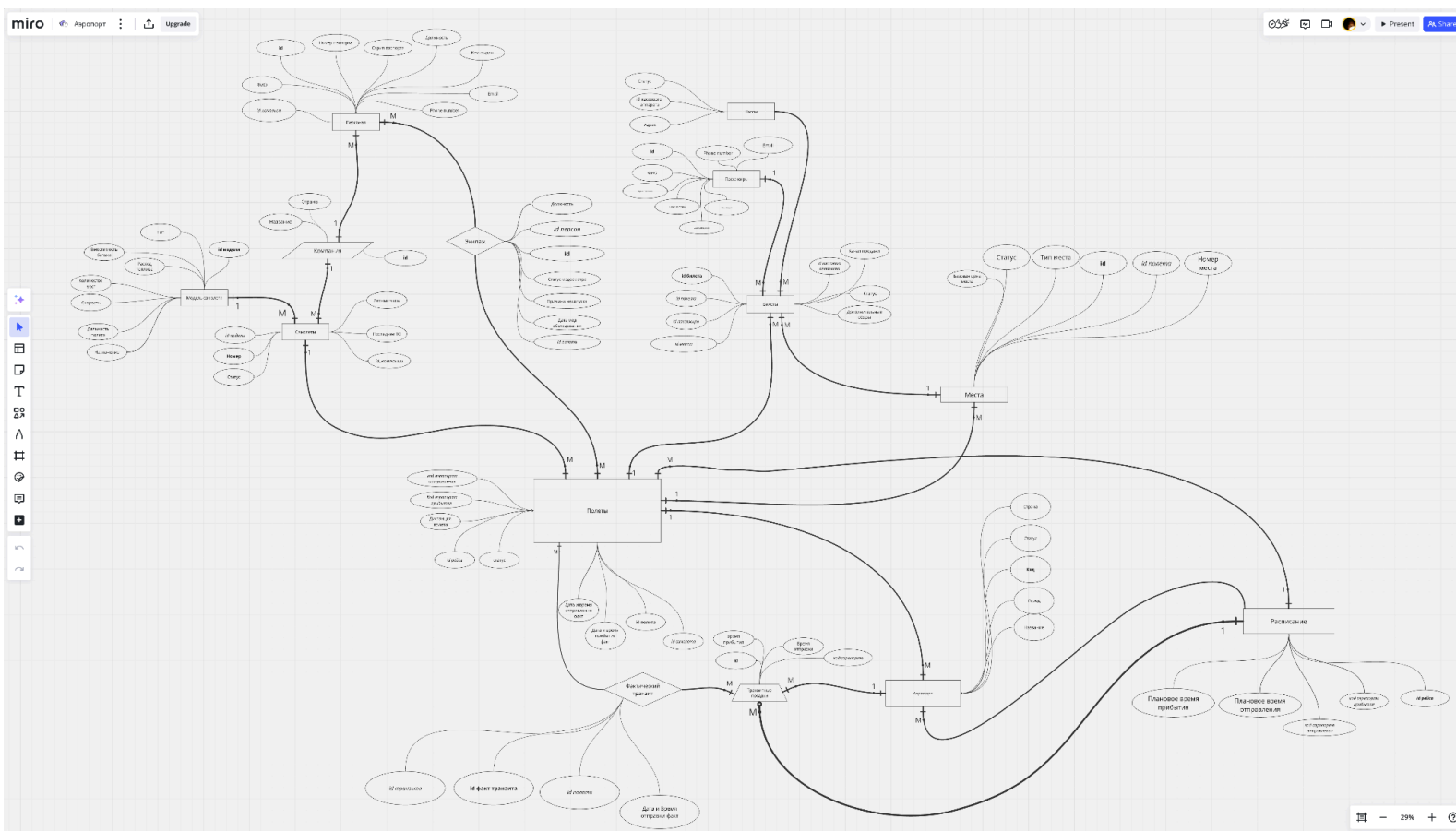


Рис. 1 - Схема инфологической модели данных БД в нотации Питера Чена-Кириллова

Также ее можно найти по вот этой ссылке

https://miro.com/app/board/uXjVKU_YAl0=?share_link_id=606424706258

На рис. 2 изображена схема инфологической модели данных БД в нотации IDEF1X.

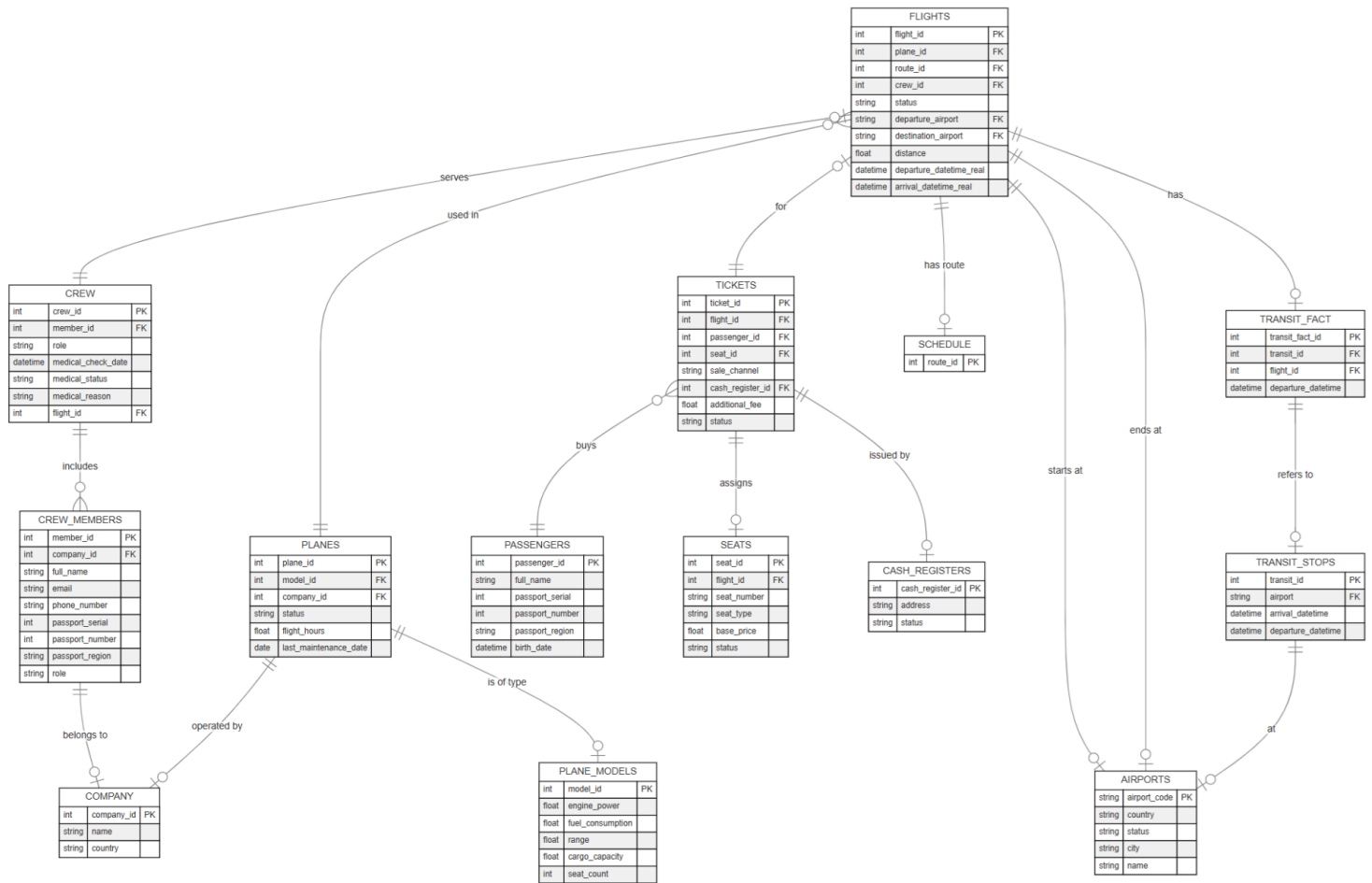


Рис. 2 - схема инфологической модели данных БД в нотации IDEF1X

В табл. 1 представлено описание атрибутов сущностей и ограничений на данные.

Таблица 1 - Описание атрибутов сущностей и ограничений на данные

Table name	Field Name	Type	Self-Attribute	Foreign key	Not Null	Integrity constraints
CREW	crew_id	int	YES	NO	YES	Unique Autoincrement
CREW	member_id	int	YES	YES	YES	Foreign key from CREW_MEMBERS(member_id);
CREW	role	string	NO	NO	YES	Not Empty
CREW	medical_check_date	datetime	NO	NO	YES	Not in future
CREW	medical_status	string	NO	NO	YES	Chosen from list (FitforDuty, UnfitforDuty, OnSickLeave, InQuarantine, Recovered, MedicalClearancePending)
CREW	medical_reason	string	NO	NO	NO	Not Empty
CREW	flight_id	int	NO	YES	YES	Foreign key from FLIGHTS(flight_id);
CREW_MEMBERS	member_id	int	YES	NO	YES	Unique Autoincrement
CREW_MEMBERS	company_id	int	NO	YES	YES	Foreign key from COMPANY(company_id);

CREW_MEMBERS	full_name	string	NO	NO	YES	Not Empty
CREW_MEMBERS	email	string	YES	NO	YES	Not Empty. Must contain a valid email address format Case-insensitive
CREW_MEMBERS	phone_number	string	YES	NO	YES	Not Empty. Must contain a valid phone number format, including optional country code
CREW_MEMBERS	passport_serial	int	NO	NO	YES	Unique in conjunction with the passport_number. Must contain only numeric characters, with a typical length constraint of 4 to 8 digits, depending on the passport-issuing country
CREW_MEMBERS	passport_number	int	NO	NO	YES	Unique in conjunction with the passport_serial. Unique in conjunction with the passport_serial. Must contain only numeric characters, with a typical length constraint of 6 to 10 digits, depending on the passport-issuing country
CREW_MEMBERS	passport_region	string	NO	NO	YES	Not Empty

CREW_MEMBERS	role	string	NO	NO	YES	Chosen from list (Pilot, CoPilot, FlightEngineer, CabinCrew, FlightAttendant, Purser, Navigator, Dispatcher, GroundSupport, MaintenanceTechnician)
COMPANY	company_id	int	YES	NO	YES	Unique Autoincrement
COMPANY	name	string	NO	NO	YES	Not Empty
COMPANY	country	string	NO	NO	YES	Not Empty
PLANES	plane_id	int	YES	NO	YES	Unique Autoincrement
PLANES	model_id	int	NO	YES	YES	Foreign key from PLANE_MODELS(model_id);
PLANES	status	string	NO	NO	YES	Chosen from list (InService, OutofService, MaintenanceRequired, InRepair, ScheduledforMaintenance, Reserved, Decommissioned)
PLANES	flight_hours	float	NO	NO	YES	>=0

PLANES	company_id	int	NO	YES	YES	Foreign key from COMPANY(company_id);
PLANES	last_maintenance_date	date	NO	NO	YES	Not in future
PLANE_MODELS	model_id	int	YES	NO	YES	Unique Autoincrement
PLANE_MODELS	engine_power	float	NO	NO	YES	0<= x <= U_FLOAT_MAX, HP
PLANE_MODELS	fuel_consumption	float	NO	NO	YES	0<= x <= U_FLOAT_MAX, Kg/miles
PLANE_MODELS	range	float	NO	NO	YES	0<= x <= U_FLOAT_MAX, miles
PLANE_MODELS	cargo_capacity	float	NO	NO	YES	0<= x <= U_FLOAT_MAX, kg
PLANE_MODELS	seat_count	int	NO	NO	YES	0<= x <= U_INT_MAX
FLIGHTS	flight_id	int	YES	NO	YES	Unique Autoincrement
FLIGHTS	plane_id	int	NO	YES	YES	Foreign key from PLANES(plane_id);
FLIGHTS	route_id	int	NO	YES	YES	Foreign key from SCHEDULE(route_id);
FLIGHTS	crew_id	int	NO	YES	YES	Foreign key from CREW(crew_id);

FLIGHTS	status	string	NO	NO	YES	Chosen from list (Scheduled, InProgress, Completed, Cancelled, Delayed, Diverted, OnTime, Boarding)
FLIGHTS	departure_air port	string	NO	NO	YES	Foreign key from AIRPORTS(airport_code)
FLIGHTS	destination_a irport	string	NO	NO	YES	Foreign key from AIRPORTS(airport_code)
FLIGHTS	distance	float	NO	NO	NO	>=0
FLIGHTS	departure_da tetime_real	datetime	NO	NO	YES	>= “1903.12.17”
FLIGHTS	arrival_dateti me_real	datetime	NO	NO	YES	> departure_datetime_real
TICKETS	ticket_id	int	YES	NO	YES	Unique Autoincrement
TICKETS	flight_id	int	NO	YES	YES	Foreign key from FLIGHTS(flight_id);
TICKETS	passenger_id	int	NO	YES	YES	Foreign key from PASSENGERS(passenger_ id);
TICKETS	seat_id	int	NO	YES	YES	Foreign key from SEATS(seat_id);
TICKETS	sale_channel	string	NO	NO	YES	Chosen from list (Online, TicketOffice, TravelAgency, MobileApp,

						Kiosk, CorporateBooking, ThirdPartyVendor, AirportCounter)
TICKETS	cash_register_id	int	NO	YES	YES	Foreign key from CASH_REGISTERS(cash_register_id);
TICKETS	additional_fee	float	NO	NO	NO	>=0
TICKETS	status	string	NO	NO	YES	Chosen from list (Available, Booked, Confirmed, Cancelled, CheckedIn, NoShow, Refunded, PendingPayment)
PASSENGERS	passenger_id	int	YES	NO	YES	Unique Autoincrement
PASSENGERS	full_name	string	NO	NO	YES	Not Empty, contains name, first name and last name
PASSENGERS	passport_serial	int	NO	NO	YES	Unique in conjunction with the passport_number. Must contain only numeric characters, with a typical length constraint of 4 to 8 digits, depending on the passport-issuing country
PASSENGER	passport_number	int	NO	NO	YES	Unique in conjunction with the passport_serial. Unique

						in conjunction with the passport_serial. Must contain only numeric characters, with a typical length constraint of 6 to 10 digits, depending on the passport-issuing country
PASSENGERS	passport_region	string	NO	NO	YES	Not Empty
PASSENGERS	birth_date	datetime	NO	NO	YES	Not in future
SEATS	seat_id	int	YES	NO	YES	Unique Autoincrement
SEATS	flight_id	int	NO	YES	YES	Foreign key from FLIGHTS(flight_id);
SEATS	seat_number	string	NO	NO	YES	Number + Letter
SEATS	seat_type	string	NO	NO	YES	Chosen from list (Economy, Business, FirstClass, PremiumEconomy, Window, Aisle, Middle, ExitRow, Bulkhead, Recliner, NonRecliner, Accessible)
SEATS	base_price	float	NO	NO	YES	0 <= x <= U_INT_MAX
SEATS	status	string	NO	NO	YES	Chosen from list (Available, Occupied,

						Reserved, Blocked, Unavailable, Checkedin, PendingAssignment)
CASH_REGISTER	cash_register_id	int	YES	NO	YES	Unique Autoincrement
CASH_REGISTER	address	string	NO	NO	YES	Not Empty
CASH_REGISTER	status	string	NO	NO	YES	Chosen from list (Open, Closed, InUse, OutofService, UnderMaintenance, Reconciled, AwaitingCash)
TRANSIT_FACT	transit_fact_id	int	YES	NO	YES	Unique Autoincrement
TRANSIT_FACT	transit_id	int	NO	YES	YES	Foreign key from TRANSIT_STOPS(transit_id);
TRANSIT_FACT	flight_id	int	NO	YES	YES	Foreign key from FLIGHTS(flight_id);
TRANSIT_FACT	departure_datetime	datetime	NO	NO	YES	>= “1903.12.17”
TRANSIT_FACT	arrival_datetime	datetime	NO	NO	YES	> departure_datetime_real
TRANSIT_STOPS	transit_id	int	YES	NO	YES	Unique Autoincrement

TRANSIT_STOPS	airport	string	NO	NO	YES	Foreign key from AIRPORTS(airport_code);
TRANSIT_STOPS	arrival_datetime	datetime	NO	NO	YES	> departure_datetime_real
TRANSIT_STOPS	departure_datetime	datetime	NO	NO	YES	>= “1903.12.17”
AIRPORTS	airport_code	string	YES	NO	YES	Unique
AIRPORTS	country	string	NO	NO	YES	Not Empty
AIRPORTS	status	string	NO	NO	YES	Chosen from list (Operational, Closed, UnderConstruction, TemporarilyClosed, LimitedOperations, RestrictedAccess, EmergencyMode)
AIRPORTS	city	string	NO	NO	YES	Not Empty
AIRPORTS	name	string	NO	NO	YES	Not Empty
SCHEDULE	route_id	int	YES	NO	YES	Unique Autoincrement
SCHEDULE	departure_airport	string	NO	NO	YES	Foreign key from AIRPORTS(airport_code);

SCHEДУ LE	destination_a irport	string	NO	NO	YES	Foreign key from AIRPORTS(airport_code);
SCHEДУ LE	departure_da tetime	datetime	NO	NO	YES	>= “1903.12.17”
SCHEДУ LE	arrival_dateti me	datetime	NO	NO	YES	> departure_datetime_real

Выводы

В ходе выполнения данного проекта по разработке ИЛМ для БД “Аэропорт” была проведен анализ, моделирование и проектирование структуры данных для образовательной системы.

На основе анализа предметной области была построена ER-диаграмма, которая наглядно демонстрировала взаимосвязи между сущностями.

На следующем этапе была разработана логическая модель базы данных с использованием нотации IDEF1X. В этой модели были описаны атрибуты сущностей, их ключи, а также связи между ними, что позволило глубже понять структуру и целостность данных.

В процессе проектирования были определены важные ограничения для обеспечения корректности данных. Уникальные ключи, ссылочная целостность и ограничения на значения атрибутов были четко сформулированы.

В заключение, выполнение данного проекта дало возможность освоить метод визуализации информации с помощью ER-диаграмм и формализации данных в нотации IDEF1X.