**КОМУНАЛЬНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД**

**«ІНСТИТУТ ПІДПРИЄМНИЦТВА «СТРАТЕГІЯ» ДОР»**

**КАФЕДРА «КОМП’ЮТЕРНІ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І МОДЕЛЮВАННЯ ЕКОНОМІКИ»**

**КУРСОВИЙ ПРОЕКТ**

з предмету «Бази даних»

на тему: *БД для АІС аеропорту*

Студента (ки) ІІ курсу П-15-51 групи

напряму підготовки «6.050103 – Програмна інженерія»

*Кочерги С.А.*

(прізвище та ініціали)

Керівник \_*доцент кафедри КІТМЕ, к.т.н. Міхальчук В.І.*

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_ Рудяков А.В.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_ Тарасюк С.В.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_ Глеб А.С.

(підпис) (прізвище та ініціали)

Жовті Води

2017 рік

**КОМУНАЛЬНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД**

**«ІНСТИТУТ ПІДПРИЄМНИЦТВА «СТРАТЕГІЯ» ДОР»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Кафедра | «Комп’ютерні та інформаційні технології і моделювання економіки» | | | | |
| Дисципліна | Бази даних | | | | |
| Напрям підготовки | Програмна інженерія | | | | |
| Курс | другий | група | П-15-51 | семестр | 3-4 |

**ЗАВДАННЯ**

|  |  |
| --- | --- |
| на курсовий проект студенту (ці) | Кочерзі Сергію Андрійовичу |
|  | |
| 1. Тема проекту | Розробка БД для АІС «Аеропорт» |
|  |  |
| 2. Строк здачі студентом закінченого проекту | 07.06.2017 р. |
| 3. Вихідні дані до проекту | пояснювальна записка друкована, файли з діаграмами та пояснювальною запискою |
| 4. Зміст пояснювальної записки (перелік питань, які підлягають розробці) | 1. Постановка задачі. |
|  | 2. Вступ. |
|  | 3. Архітектура та інформаційне забезпечення АІС. |
|  | 3.1 Структура ІС, її призначення. |
|  | 3.2 Діаграми потоків даних. |
|  | 3.3 Основні процеси. Задачі по підрозділам. |
|  | 3.4 Матриці подій для кожного робочого місця. |
|  | 3.5 Бізнес-правила та обмеження цілісності. |
|  | 3.6 Життєві цикли основних інформаційних об’єктів. |
|  | 3.7 Компонентна модель: структура об’єктів, зв’язки, методи. |
|  | 4 Програмна реалізація системи |
|  | 4.1 Структура програмного забезпечення |
|  | 4.2 Опис програми (Текст sql-сценаріїв, тригерів, зберігаємих процедур). |
|  | 5. Керівництво користувача. |
|  | 6. Мінімальні вимоги до апаратного забезпечення. |
|  | 7. Випробування роботи системи (екранні копії роботи програми). |
|  | 8. Висновки. |
|  | 9. Перелік використаних джерел. |
| 5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) | 1. ER-діаграма логічна  2. ER-діаграма фізична |
| 6. Дата видачі завдання | 12.09.2016 |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ з/п** | **Назва етапів курсового проекту (роботи)** | **Строк виконання етапів роботи** | **Примітка** |
|  | Вибір теми курсової роботи. Опрацювання відповідної літератури. Оформлення листа Завдання. | 12.09.2016 |  |
|  | Аналіз постановки задачі. | 01.10.2016 |  |
|  | Аналіз предметної галузі. | 10.10.2016 |  |
|  | Бізнес правила + Життєві цикли + Компонентна модель | 17.11.2016 |  |
|  | ER-діаграма логічна. | 05.12.2016 |  |
|  | ER-діаграма фізична. | 19.12.2016 |  |
|  | Проміжковий звіт | 25.12.2016 |  |
|  | Вивчення середовища програмної реалізації системи. | 23.02.2017 |  |
|  | \*Програмна реалізація інтерфейсу користувача. | 27.03.2017 | За умови реалізації |
|  | Реалізація засобів адміністрування системи. | 24.04.2017 |  |
|  | Реалізація звітів та бізнес-правил. | 22.05.2017 |  |
|  | Заключне тестування програми. | 17.05.2017 |  |
|  | Аналіз результатів. | 22.05.2017 |  |
|  | Оформлення пояснювальної записки. | 23.05.2017 |  |
|  | Захист та демонстрація курсової роботи. | 07.06.2017 – «А»  08.06.2017 –«В,С»  09.06.2017 – «D, E» |  |

|  |  |  |
| --- | --- | --- |
| Студент: |  | Кочерга Сергій Андрійович |
|  |  |  |
| Керівник: |  | Міхальчук Віктор Ілліч |
|  | (підпис) | (П.І.Б.) |
| Дата: |  | |

Оціночний лист курсового проекту

студента Кочерги Сергія Андрійовича гр. П-15-51

**Оцінка пояснювальної записки**

(кількість балів за окремий вид робіт встановлюється кафедрою, виходячи із специфіки дисципліни, за якою виконується дана робота – відповідно до критеріїв оцінювання)

**Зміст пояснювальної записки, в тому числі: до 50 балів**

Постановка задачі (опис інфологічної моделі): до 10 балів

Логічність структури роботи та її відповідність обраній темі \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Обґрунтування актуальності теми роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(бали)

Логічне проектування БД: до 20 балів

Чіткість визначення об’єкту та предмету дослідження \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Якість та повнота проведеного проектування \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(бали)

Фізичне проектування та реалізація БД: до 20 балів

Повнота реалізації сутностей \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_(бали)

Повнота висновків \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_(бали)

Обґрунтованість авторських рішень, пропозицій (результатів) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

**Оформлення пояснювальної записки: до 20 балів**

Наочність подання результатів аналізу (таблиці, графіки, малюнки) \_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Наявність та коректність посилань на літературні джерела \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Склад та структура списку використаних джерел \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Наявність та якість додатків \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Зовнішнє оформлення роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

**Своєчасність виконання етапів роботи: до 5 балів**

Своєчасність та самостійність роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Творчість та науковість розв’язання теми \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (бали)

Підсумкова оцінка пояснювальної записки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(до 75)

**Оцінка процедури захисту**

Оцінка захисту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (до 25)

(рекомендується оцінювати доповідь, стиль мови та змістовність відповідей на запитання окремо)

Науковий керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Зміст

[Перелік скорочень та ключових слів 7](#_Toc484589879)

[Вступ 8](#_Toc484589880)

[1 Постановка задачі 10](#_Toc484589881)

[2 проектування БД 11](#_Toc484589882)

[2.1 Організаційні основи аеропорту та аналіз функціонування 11](#_Toc484589883)

[2.1.1 Питання про маршрути 12](#_Toc484589884)

[2.1.2 Основні сутності 13](#_Toc484589885)

[2.1.3 Життєвий цикл клієнта 13](#_Toc484589886)

[2.2 Інфологічне проектування 13](#_Toc484589887)

[2.2.1 Визначення первісного набору відношень 14](#_Toc484589888)

[2.3 Даталогічне проектування 18](#_Toc484589889)

[2.3.1 Виділення сутностей та зв’язків між ними 18](#_Toc484589890)

[2.3.2 Побудова узагальненої ER-діаграми 20](#_Toc484589891)

[2.4 Фізичне проектування 21](#_Toc484589892)

[3 Фізична реалізація 27](#_Toc484589893)

[3.1 Таблиці та зв’язки між ними 27](#_Toc484589894)

[3.2 Тестові запити 29](#_Toc484589895)

[3.2.1 Запит1 30](#_Toc484589896)

[3.2.2 Запит 2 31](#_Toc484589897)

[3.2.3 Запит 3 32](#_Toc484589898)

[3.2.4 Запит 4 32](#_Toc484589899)

[3.2.5 Запит 5 33](#_Toc484589900)

[3.2.6 Запит 6 33](#_Toc484589901)

[3.2.7 Запит 7 34](#_Toc484589902)

[3.2.8 Запит 8 34](#_Toc484589903)

[3.2.9 Запит 9 35](#_Toc484589904)

[3.2.10 Запит 10 35](#_Toc484589905)

[3.2.11 Запит 11 36](#_Toc484589906)

[3.2.12 Запит 12 36](#_Toc484589907)

[Висновки 38](#_Toc484589908)

[Перелік Використаних джерел 39](#_Toc484589909)

[Додаток А er-діаграма бд 40](#_Toc484589910)

[Додаток Б діаграма зв’язків між таблицями бд 41](#_Toc484589911)

[Додаток В Дамп бази даних 42](#_Toc484589912)

Перелік скорочень та ключових слів

АІС – автоматична інформаційна система, сторінка 13.

БД – база даних, сторінка 7.

SQL –  (англ. Structured query language — мова структурованих запитів) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних, сторінка 7.

Лог – файл (таблиця) реєстрації квитків, сторінка 14.

СУБД – система управління базою даних, сторінка 13.

ПІБ – прізвище, ім’я, по-батькові, сторінка 12.

ER-діаграма – це модель «сутність-зв'язок» (ER-модель) (англ. Entity-relationship model або entity-relationship diagram) — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків, сторінка 20.

Вступ

Курсовий проект з предмету «Бази даних» призначений для:

* вдосконалення навичок з аналізу предметного середовища з метою побудови еквівалентних інформаційних моделей з використанням відповідного ПЗ;
* засвоєння методології поетапного проектування реляційних моделей баз даних з використанням інформаційних діаграм;
* практичного застосування вивчених протягом курсу запитів мовою SQL.

Предметна галузь дійсного курсового проекту – аеропорт (здійснює перевезення, шляхом перельоту на літаку, людей, техніки, різних товарів та вантажу між містами, країнами у відповідне місце призначення, де є аеропорт з необхідними умовами обслуговування літака. Перельоти здійснюються в різні пункти призначення, куди саме необхідно доставити пасажирів та вантаж).

Актуальність курсового проекту обумовлена необхідністю набуття автором теоретичних та практичних знань в галузі проектування та фізичної реалізації БД засобами обраного інструментарію.

Ціль створення проекту – отримати працездатну БД, що призначена для реалізації інформаційної моделі аеропорту у відповідності до навчального завдання.

Об’єктом дослідження було визначено інформаційні потоки аеропорту, що необхідно враховувати під час створення інформаційної моделі. Всі умови, що не були визначені навчальним завданням, були до визначені автором у межах логіки проекту, або ігнорувались як несуттєві.

Предметом дослідження стали процеси збору, обробки та передачі інформації про бізнес-середовище аеропортів:

* організацію рейсів;
* обслуговування літаків;
* рішення кадрових питань;
* створення та організація обслуговуючих бригад;
* інші питання предметної галузі.

Таким чином було проведено дослідження предметної галузі та створено проект реляційної моделі БД для опису бізнес-середовища «Аеропорт». Створені в ході проектування запити призначені для роботи з проектованою БД у відповідності до бізнес-логіки – повідомляти значення, отримані від СУБД, які бажає отримати користувач БД. Було також визначено основні функції програм, які використовувались для розробки БД.

Пояснювальну записку з курсового проектування укладено у відповідності до вимог [1, 2] з метою пояснення виконаних дій та прийнятих в ході проектування рішень.

1. Постановка задачі

Необхідно створити проект бази даних для автоматизованої інформаційної системи аеропорту.

БД автоматизованої інформаційної системи призначена для використання в роботі аеропорту. Її призначенням є підвищення:

* якості (повноти, достовірності, своєчасності, узгодженості) службової інформації аеропорту, що в свою чергу дозволить більш раціонально використовувати наявні ресурси льотної компанії;
* якості обслуговування клієнтів цього аеропорту за рахунок зменшення часу реєстрації пасажирів на рейси та зменшення кількості квитків (виданих за одиницю часу) з невірно вказаною інформацією;
* підвищення технічного стану літаків за рахунок своєчасного техогляду та оновлення інформації;
* швидке вирішення питань з багажем;
* якість оновлення інформації про персонал аеропорту, рейс та літаків;
* швидке планування рейсів;
* ефективний розподіл праці між персоналом;
* рівня безпечності зберігання даних.

БД системи повинна допомагати реалізувати функції системи:

* реєстрація клієнтів;
* реєстрація багажу клієнтів;
* реєстрація рейсів;
* продаж квитків на заплановані рейси;
* зберігання та обробка інформації для звітів;
* визначення процентного відношення незатребуваних місць;
* технічний стан літаків;
* стан рейсу;
* повна інформація про персонал працівників аеропорту.

1. проектування БД
   1. Організаційні основи аеропорту та аналіз функціонування

Процес створення оптимальних маршрутів є досить складним і не автоматизується в даному проекті, тому далі його розгляд не є доречним.

Підставою для початку продажу квитків є реєстрація рейсу, який характеризується:

* унікальним номером рейсу;
* маршрутом;
* датою і часом відправлення з початкового пункту;
* датою і часом прибуття в кінцевий пункт;
* категорією;
* затримкою та часом затримки;
* часом польоту;
* статусом;
* ціною за квиток;
* транспортним засобом, що призначений на виконання рейсу.

Від останнього залежить:

* кількість пасажирських місць;
* спроможність подолати необхідну відстань;
* час польоту;
* технічний стан.

Для зазначення маршруту диспетчери вказують аеропорти відправки та прибуття, що мають:

* унікальний номер;
* назву;
* географічні координати.

Назва аеропорту зазвичай відповідає населеному пункту, в якому він знаходиться.

Кожен проданий квиток має наступні реквізити:

* унікальний номер квитка;
* ПІБ клієнта, що його придбав;
* номер рейсу;
* пункт відправлення;
* пункт призначення;
* ціна за переліт.

Є недопустимою ситуація, коли кількість пасажирів перевищує кількість сидячих місць у рейсі. Реєстрація клієнтів на рейс обов’язково завершується, коли настає дата і час відправлення літака з початкового пункту.

Для забезпечення зв’язку з клієнтами аеропорт зберігає наступну їхню персональну інформацію:

* унікальний номер клієнта в БД;
* ПІБ;
* стать;
* номер рейсу;
* номер місця;
* вік.
  + 1. Питання про маршрути

Використання географічних координат для аеропортів є прийнятним для визначення довжин маршрутів, оскільки у повітрі літак долає меншу відстань, ніж це робив би транспорт на землі. Один з варіантів покращення перельоту – покращення планування маршрутів для літаків, заздалегідь точно відомий стан погоди, якісне обслуговування літака, якщо необхідно - дозаправка. Але дозаправка вимагає більше часу на рейс, оскільки літаку необхідно здійснювати посадку в проміжних аеропортах, тому для рейсу буде правильним підбирати таку модель літака, якій не потрібна буде дозаправка.

При цьому дійсно час перельоту не втрачається. Розглянута пропозиція може обійтись у великі кошти, оскільки такі моделі літаків коштують дуже багато, як і їх обслуговування. Але з часом всі витрати окупляться, оскільки клієнти будуть задоволені сервісом аеропорту і рекомендуватимуть його іншим.

* + 1. Основні сутності

Враховуючи описані вище організаційні основи аеропорту можна виділити наступні основні об’єкти (сутності) предметної області:

* клієнт;
* багаж;
* квиток(та лог квитків);
* рейс;
* персонал;
* посада;
* професія;
* літак;
* аеропорт.
  + 1. Життєвий цикл клієнта

Реєстрація на рейс клієнтом (один з основних бізнес процесів АІС «Аеропорт») відбувається за схемою (дивись рисунок 2.1).



* + - 1. Життєвий цикл клієнта
  1. Інфологічне проектування

Концептуальне (інфологічне) проектування бази даних – процедура побудови інформаційної моделі даних, що не залежить від будь-яких фізичних умов реалізації, а саме від: типу СУБД, складу програм додатку, мови програмування, обчислювальної платформи, тощо. Ця модель має якомога ширше описувати предметне середовище.

Визначимо сутності в предметній області:

* персонал;
* обслуговуючі бригади;
* стать;
* професії;
* льотна бригада;
* зарплата;
* посада;
* аеропорти;
* літаки;
* рейс
* категорії рейсів;
* причини затримки;
* моделі літаків;
* пасажири;
* квитки;
* лог квитків;
* багаж.
  + 1. Визначення первісного набору відношень

На основі виділених сутностей визначаємо первісний набір відношень (дивись рисунок 2.10). Зв’язки між відношеннями реалізуємо за допомогою зовнішніх ключів, назви яких співпадають з назвами первісних ключів, на які виконуються посилання.

Для інших сутностей характер первісних наборів відношень формувався аналогічним чином, тож, для спрощення в пояснювальній записці наведено лише один приклад, який автор вважає характерним.

Відношення в межах сутностей є зрозумілими та такими, що відповідають бізнес-логіці, а відтак нормалізації не потребують.



* + - 1. Первісний набір відношень

Перерахуємо атрибути сутностей, тобто яку інформацію про сутності будемо зберігати в базі даних (дивись таблицю 2.1).

Таблиця 2.1 Атрибути та їх сутності

| **Сутності** | **Атрибути сутностей** |
| --- | --- |
| Персонал | код персоналу  прізвище  ім’я  по батькові  стать  вік  стаж роботи  медогляд  адреса  діти  кількість дітей  зарплата  посада  професія  № обслуговуючої бригади  № льотної бригади  код рейсу |
| Обслуговуючі бригади | код обслуговуючої бригади  номер бригади |
| Стать | код статі  назва статі |
| Професії | код професії  професія |
| Льотна бригада | код льотної бригади  номер льотної бригади |
| Зарплата | код зарплати  розмір |
| Посада | код посади  посада |
| Аеропорти | код аеропорту  назва аеропорту  місце розташування |
| Літаки | бортовий номер  вік експлуатації  техогляд  дата техогляду  кількість ремонтів  дата останнього ремонту  кількість рейсів до ремонту  модель  № льотної бригади  № обслуговуючої бригади  аеропорт приписки  дата прибуття в аеропорт |
| Рейс | номер рейсу  бортовий номер літака  категорія  номер квитка  час вильоту  час посадки  час польоту  дата рейсу  затримка  причина затримки  час затримки  аеропорт відправки  аеропорт прибуття  статус |
| Моделі літаків | код моделі  модель літака  кількість пасажирських місць |
| Причини затримки | код затримки  назва затримки |
| Категорії рейсів | код категорії  назва категорії |
| Пасажири | серія та номер паспорту  ПІБ  стать  вік  № рейсу  місце |
| Квитки | код вартості квитка  вартість квитка в доларах |
| Лог квитків | код логу  код рейсу  код літака  ціна  код власника  аеропорт відправки  аеропорт прибуття  № місця  дата покупки  статус |
| Багаж | код багажу  вага багажу(кг)  власник |

* 1. Даталогічне проектування

Фаза логічного (даталогічного) проектування БД полягає в перетворенні концептуальної моделі даних в логічну модель, враховуючи тип СУБД. Логічна модель даних є джерелом інформації для фізичного проектування. Цей етап передбачає всебічний аналіз різноманітних аспектів роботи з даними, що має виключно важливе значення для ефективного вирішення поставлених задач.

Цільова СУБД в даному проекті – реляційна.

* + 1. Виділення сутностей та зв’язків між ними

На основі інфологічної моделі, враховуючи організаційні основи аеропорту, можна виділити наступні сутності та їх атрибути (ключові атрибути підкреслено):

* персонал: код персоналу, прізвище, ім’я, по батькові, стать, вік, стаж роботи, медогляд, адреса, діти, кількість дітей, зарплата, посада, професія, № обслуговуючої бригади, № льотної бригади, код рейсу;
* літак: бортовий номер, час прибуття в аеропорт, вік експлуатації, техогляд, дата техогляду, кількість ремонтів, дата останнього ремонту, кількість рейсів до ремонту, модель, № льотної бригади, № обслуговуючої бригади, аеропорт приписки;
* аеропорт: код аеропорту, назва аеропорту, місце знаходження;
* рейс: номер рейсу, бортовий номер літака, номер квитка, час вильоту, час посадки, час польоту, дата рейсу, затримка, час затримки, аеропорт відправки, аеропорт прибуття, статус;
* квитки: код квитка, вартість;
* лог квитків: код логу, номер рейсу, код літака, ціна, код власника, кількість проданих, аеропорт відправки, аеропорт прибуття, № місця, дата покупки, статус;
* пасажир: серія та номер паспорту, ПІБ, вік, стать, № рейсу, місце;
* багаж: код багажу, вага(кг), власник.

Атрибути кожної сутності є простими. Слід зазначити, що дату і час відправлення, що зазначається в квитку, можна розрахувати, виходячи з даних про рейс та пункт відправлення. Ціну квитку можна обчислити за даними про рейс, пункт відправлення та пункт призначення (що зазначені в квитку). № місця в квитку бажано перевіряти на сумісність з іншими проданими квитками. Пункт відправлення і пункт призначення у квитку повинні мати сенс для рейсу, на який придбано квиток. Дату і час прибуття в кінцевий пункт рейсу також можна розрахувати.

Зв’язки між сутностями:

* «персонал» обслуговує «літак»;



* + - 1. Зв’язок між сутностями «персонал» і «літак»
* «літак» приписаний до «аеропорт»;

**

* + - 1. Зв’язок між сутностями «літак» і «аеропорт»
* «аеропорт» організовує «рейс»;

**

* + - 1. Зв’язок між сутностями «аеропорт» і «рейс»
* на «рейс» продано «квитки», при чому характер зв’язку не обов’язковий, так як відображення факту реєстрації рейсу не означає, що хоч один квиток на нього буде продано;



* + - 1. Зв’язок між сутностями «рейс» і «квитки»
* «квитки» мають «лог квитків»;



* + - 1. Зв’язок між сутностями «квитки» і «лог квитків»
* «пасажир» купує «квитки»;



* + - 1. Зв’язок між сутностями «пасажир» і «квитки»
* «пасажир» має «багаж»;



* + - 1. Зв’язок між сутностями «пасажир» і «багаж»

За множинністю всі зв’язки відносяться до типу 1:М і є бінарними. Жоден зв’язок не має атрибутів.

* + 1. Побудова узагальненої ER-діаграми

Діаграма ER-типу (дивись додаток А) відображує загальну структуру логічної моделі даних. Підкреслені атрибути є ключовими, жирним шрифтом зазначені зовнішні ключі, виділені темно-сірим кольором атрибути можливо обчислити на основі інших даних, виділені світло-сірим кольором атрибути потребують (бажано) додаткових обмежень цілісності даних.

* 1. Фізичне проектування

Етап фізичного проектування передбачає створення таблиць БД на основі логічного проекту. Кожній сутності з логічного проекту має відповідати таблиця у фізичному проекті, таким чином необхідно створити наступні таблиці:

* персонал;
* обслуговуючі бригади;
* стать;
* професії;
* льотна бригада;
* зарплата;
* посада;
* аеропорти;
* літаки;
* рейс;
* категорії рейсів;
* причини затримки;
* моделі літаків;
* пасажири;
* квитки;
* лог квитків;
* багаж.

Таблиці 2.2 – 2.18 відповідають логічному проекту.

Таблиця 2.2 Сутність «Аеропорт»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код аеропорту | Лічильник (11), ціле | Персональний код аеропорту |
| Назва | Текстовий (50) | Назва аеропорту |
| Місце розташування | Текстовий (50) | Місце розташування аеропорту |

Таблиця 2.3 Сутність «Багаж»

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| \*Код багажу | Лічильник (11), ціле | Персональній код багажу |
| Вага багажу (кг) | Числовий, ціле (11) | Вага багажу |
| Власник | Текстовий (50) | Зовнішній ключ таблиці «Пасажир» |

Таблиця 2.4 Сутність «Зарплата»

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| \*Код зарплати | Лічильник (11), ціле | Персональній код зарплати |
| Розмір | Грошовий (50) | Розмір зарплати |

Таблиця 2.5 Сутність «Квитки»

|  |  |  |
| --- | --- | --- |
| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| \*Код квитка | Лічильник (11), ціле | Персональній код квитка |
| Вартість квитка | Числовий, ціле | Вартість квитка |

Таблиця 2.6. Сутність «Літак»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Бортовий номер | Лічильник (11), ціле | Персональний код літака |
| Вік експлуатації | Числовий, ціле (11) | Вік експлуатації літака |
| Техогляд | Логічний | Відмітка про техогляд літака |
| Дата техогляду | Дата/час | Дата техогляду літака |
| Кількість ремонтів | Числовий, ціле (11) | Кількість ремонтів літака |
| Дата останнього техогляду | Дата/час | Дата останнього техогляду літака |
| Кількість рейсів до ремонту | Числовий, ціле (11) | Кількість рейсів літака до ремонту |
| Модель | Текстовий (50) | Зовнішній ключ таблиці «Моделі літаків» |
| № льотної бригади | Числовий, ціле (11) | Зовнішній ключ таблиці «Льотна бригада» |
| № обслуговуючої бригади | Числовий, ціле (11) | Зовнішній ключ таблиці «Обслуговуючі бригади» |
| Аеропорт приписки | Текстовий (50) | Зовнішній ключ таблиці «Аеропорт» |
| Час надходження в аеропорт | Дата/час | Дата надходження літака до аеропорту приписки |

Таблиця 2.7 Сутність «Льотна бригада»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код льотної бригади | Лічильник (11), ціле | Персональний код літака |
| № льотної бригади | Числовий, ціле (11) | Номер льотної бригади |

Таблиця 2.8 Сутність «Професії»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код професії | Лічильник (11), ціле | Персональний код професії |
| Професія | Текстовий (50) | Назва професії |

Таблиця 2.9 Сутність «Рейс»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Номер рейсу | Лічильник (11), ціле | Персональний код рейсу |
| Бортовий номер літака | Числовий, ціле (11) | Зовнішній ключ таблиці «Літак» |
| Категорія рейсу | Текстовий(50) | Зовнішній ключ таблиці «Категорії рейсів» |
| № квитка | Числовий, ціле (11) | Зовнішній ключ таблиці «Квитки» |
| Час вильоту | Дата/час | Час вильоту літака |
| Час посадки | Дата/час | Час посадки літака |
| Час польоту | Дата/час | Час польоту літака |
| Дата рейсу | Дата/час | Дата рейсу |
| Затримка | Логічний | Затримка літака |
| Причина затримки | Текстовий(50) | Зовнішній ключ таблиці «Причини затримки» |
| Час затримки | Дата/час | Час затримки |
| Аеропорт відправки | Текстовий (50) | Зовнішній ключ таблиці «Аеропорт» |
| Аеропорт прибуття | Текстовий (50) | Зовнішній ключ таблиці «Аеропорт» |
| Статус | Текстовий(50) | Статус рейсу в даний момент часу |

Таблиця 2.10 Сутність «Стать»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код статі | Лічильник (11), ціле | Персональний код статі |
| Стать | Текстовий (50) | Зовнішній ключ таблиці «Стать» |

Таблиця 2.11 Сутність «Категорії рейсів»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код категорії | Лічильник (11), ціле | Персональний код категорії |
| Назва категорії | Текстовий (50) | Назва категорії |

Таблиця 2.12 Сутність «Причини затримки»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код затримки | Лічильник (11), ціле | Персональний код затримки |
| Назва затримки | Текстовий (50) | Назва затримки |

Таблиця 2.13 Сутність «Обслуговуючі бригади»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код обслуговуючої бригади | Лічильник (11), ціле | Персональний код обслуговуючої бригади |
| № обслуговуючої бригади | Числовий, ціле (11) | Номер обслуговуючої бригади |

Таблиця 2.14 Сутність «Пасажир»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код пасажира | Лічильник (11), ціле | Персональний код пасажира |
| Номер паспорту | Текстовий (50) | Паспорт пасажира |
| ПІБ | Текстовий (50) | ПІБ пасажира |
| Стать | Текстовий (50) | Зовнішній ключ таблиці «Стать» |
| Вік | Числовий, ціле (11) | Вік пасажира |
| № Рейсу | Числовий, ціле (11) | Зовнішній ключ таблиці «Рейс» |
| Місце | Числовий, ціле (11) | Місце в літаку |

Таблиця 2.15 Сутність «Персонал»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код персоналу | Лічильник (11), ціле | Персональний код персоналу |
| Прізвище | Текстовий (50) | Прізвища персоналу |
| Ім’я | Текстовий (50) | Ім’я персоналу |
| По-батькові | Текстовий (50) | По-батькові персоналу |
| Стать | Текстовий (50) | Зовнішній ключ таблиці «Стать» |
| Вік | Числовий, ціле (11) | Вік персоналу |
| Стаж роботи | Числовий, ціле (11) | Стаж персоналу |
| Медогляд | Дата/час | Дата останнього медогляду |
| Адреса | Текстовий (50) | Адреса персоналу |
| Діти | Логічний | Діти персоналу |
| Кількість дітей | Числовий, ціле (11) | Кількість дітей |
| Зарплата | Грошовий | Зовнішній ключ таблиці «Зарплата» |
| Посада | Текстовий (50) | Зовнішній ключ таблиці «Посада» |
| Професія | Текстовий (50) | Зовнішній ключ таблиці «Професія» |
| № обслуговуючої бригади | Числовий, ціле (11) | Зовнішній ключ таблиці «Обслуговуючі бригади» |
| № льотної бригади | Числовий, ціле (11) | Зовнішній ключ таблиці «Льотна бригада» |
| Номер рейсу | Числовий, ціле(11) | Зовнішній ключ таблиці «Рейси» |

Таблиця 2.16 Сутність «Моделі літаків»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код моделі | Лічильник (11), ціле | Персональний код моделі |
| Модель літака | Текстовий (50) | Назва моделі літака |
| Кількість пасажирських місць | Числовий, ціле(50) | Кількість пасажирських місць в літаку |

Таблиця 2.17 Сутність «Посада»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код посади | Лічильник (11), ціле | Персональний код посади |
| Посада | Текстовий (50) | Назва посади |

Таблиця 2.18 Сутність «Лог квитків»

| **Назва поля** | **Тип даних (розмір)** | **Примітка** |
| --- | --- | --- |
| \*Код логу | Лічильник (11), ціле | Персональний код логу квитка |
| Номер рейсу | Числовий, ціле (11) | Зовнішній ключ таблиці «Рейс» |
| Номер літака | Числовий, ціле (11) | Зовнішній ключ таблиці «Літак» |
| Ціна | Грошовий | Зовнішній ключ таблиці «Квиток» |
| Код пасажира | Числовий, ціле (11) | Зовнішній ключ таблиці «Пасажир» |
| К-ть продаж | Числовий, ціле (11) | К-ть продаж |
| Аеропорт відправки | Текстовий(50) | Зовнішній ключ таблиці «Аеропорт» |
| Аеропорт прибуття | Текстовий(50) | Зовнішній ключ таблиці «Аеропорт» |
| № місця | Числовий, ціле (11) | Номер місця в літаку |
| Дата покупки | Дата/час | Дата покупки квитка |
| Статус | Текстовий(50) | Статус квитка в даний момент часу |

Загальна структура спроектованої БД у вигляді діаграми зв’язків між таблицями наведена в Додатку Б (використано графічний клієнт MySQL Workbench 6.3).

1. Фізична реалізація

Фізичне проектування БД – процес створення опису конкретної реалізації бази даних, розміщеній у вторинній пам’яті. Передбачає опис структури зберігання даних й методів, призначених для здійснення найбільш ефективного доступу до інформації. Фізичне проектування обов’язково враховує всі особливості обраної СУБД. Між фазами фізичного та логічного проектування завжди є обернений зв’язок, оскільки рішення, прийняті на етапі фізичного проектування для покращення продуктивності розроблюваної системи, можуть потребувати деякого перегляду логічної моделі даних.

Враховуючи вимоги до технічних та програмних засобів середовищем розробки було обрано ноутбук Packard bell EasyNote ENTG81BA (мій робочий комп’ютер) з наступними характеристиками:

CPU: Intel® Celeron® Processor N3050(1.6 GHz), RAM: 4 GB DDR3 L Memory, HDD: 500 GB. Встановлена операційна система: Windows 10 Pro (x64).

В якості програмного середовища розробки взято Denwer 3 Base, до складу якого входять:

* СУБД MySQL 5.5;
* графічний клієнт MySQL Workbench 6.3 CE.

Основним графічним клієнтом було обрано HeidiSQL 9.1.0.4867 Portable.

* 1. Таблиці та зв’язки між ними

Перш за все необхідно створити базу даних та зробити її поточною (це можливо зробити лише в тому випадку, якщо на це є відповідні права, що надаються адміністратором БД).

CREATE DATABASE aeroport;

USE aeroport;

Далі створюємо таблиці, що відповідають сутностям сильного типу.

CREATE TABLE IF NOT EXISTS `passegers` (

`code\_passg` int(11) NOT NULL AUTO\_INCREMENT, /\*тобто в цьому полі заборонено використання значення NULL; при додаванні чергового запису в таблицю clients MySQL автоматично запише в це поле номер, на одиницю більше попереднього\*/

`ser\_numb\_passp` char(10) DEFAULT NOT NULL – поле, яке обов’язкове для заповнення даними,

`Full name` char(50) DEFAULT NOT NULL,

`gender` char(50) DEFAULT NULL,

`age` int(11) DEFAULT NULL,

`№\_of\_flight` int(11) DEFAULT NULL,

`№\_of\_placement` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_passg`),--первинний ключ для пасажирів

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `airports` (

`code\_airport` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_airport` char(255) DEFAULT NULL,

`airport\_placement` char(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`code\_airport`),

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `flights` (

`code\_flight` int(255) NOT NULL AUTO\_INCREMENT,

`bort\_numb\_of\_plane` int(255) DEFAULT NULL,

`category` char(50) DEFAULT NULL,

`code\_of\_ticket` int(11) DEFAULT NULL,

`take\_off\_time` time DEFAULT '00:00:00',

`boarding\_time` time DEFAULT '00:00:00',

`time\_of\_flighting` time DEFAULT '00:00:00',

`date\_of\_flighting` date DEFAULT NULL,

`delay` set('Yes','No') DEFAULT NULL,

`reason\_of\_delay` char(50) DEFAULT NULL,

`time\_of\_delay` time NOT NULL DEFAULT '00:00:00',

`send\_airport` char(50) DEFAULT NULL,

`coming\_airport` char(50) DEFAULT NULL,

`status\_\_` set('Fulfilled','In process','Canceled') DEFAULT NULL,

PRIMARY KEY (`code\_flight`),

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

Наступний крок – створення «більш слабких» таблиць, по відношенню до вже створених. У цих таблицях необхідно визначити зовнішні ключі, які реалізують зв’язки між сутностями логічної моделі даних.

CREATE TABLE IF NOT EXISTS `gender` (

`code\_gender` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_gender` char(50) DEFAULT NULL,

PRIMARY KEY (`code\_gender`),

KEY `name\_of\_gender` (`name\_of\_gender`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8 COMMENT='pol';

CREATE TABLE IF NOT EXISTS `planes\_models` (

`code\_model` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_model` char(50) NOT NULL DEFAULT '0',

`quantity\_of\_passg\_places` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`code\_model`),

KEY `name\_of\_model` (`name\_of\_model`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

В усіх наступних створюваних таблицях застосовуються аналогічні тригери при визначенні зовнішніх ключів.

* 1. Тестові запити

Для того, щоб перевірити працездатність запитів, необхідно спочатку заповнити таблиці тестовими даними, за допомогою оператора INSERT.

INSERT INTO `airports` (`code\_airport`, `name\_of\_airport`, `airport\_placement`) VALUES

(1, 'Algera Phitilia', 'Italy'),

(2, 'Amsterdam Spiphol', 'Netherlands'),

(3, 'Bryansk', 'Russia'),

(4, 'Vilnews', 'Lithuania'),

(5, 'Bordo Merenyak', 'France'),

(6, 'Luxembourg Phindel', 'Luxembourg'),

(7, 'Aberdin Dais', 'England'),

(8, 'Berlin Bradenbourg', 'Germany'),

(9, 'Hong Kong', 'China'),

(10, 'Dnipropetrovsk', 'Ukraine'),

(11, 'Wece', 'Germany');

INSERT INTO `passegers` (`code\_passg`, `ser\_numb\_passp`, `Full name`, `gender`, `age`, `№\_of\_flight`, `№\_of\_placement`) VALUES

(1, 'AO 221111', 'Kamyaniy Sergey Vladimirovich', 'Male', 24, 5, 44),

(2, 'AO 234242', 'Loboda Bogdan Olexandrovich', 'Male', 34, 6, 156),

(3, 'AO 328481', 'Polyakova Yuliya Mikolaevna', 'Female', 22, 2, 55),

(4, 'AO 332244', 'Elnikova Valeria Vitaliivna', 'Female', 33, 1, 56),

(5, 'AO 345677', 'Bondarenko Dmytro Panasovich', 'Male', 45, 3, 23),

(6, 'AO 356905', 'Yakovenko Volodimir Yuriyovich', 'Male', 27, 7, 12),

(7, 'AO 423842', 'Cherkaschenko Nikita Yuriyovich', 'Male', 54, 8, 99),

(8, 'AO 432824', 'Zakharchenko Yuri Anatoliyovich', 'Male', 22, 9, 44),

(9, 'AO 567890', 'Zelenskaya Anastasiya Sergiivna', 'Female', 19, 4, 77),

(10, 'AO 764583', 'Andreev Olexandr Vitalievich', 'Male', 20, 10, 123);

INSERT INTO `flights` (`code\_flight`, `bort\_numb\_of\_plane`, `category`, `code\_of\_ticket`, `take\_off\_time`, `boarding\_time`, `time\_of\_flighting`, `date\_of\_flighting`, `delay`, `reason\_of\_delay`, `time\_of\_delay`, `send\_airport`, `coming\_airport`, `status\_\_`) VALUES

(1, 1, 'international', 1, '10:24:32', '13:44:52', '03:20:20', '2016-09-10', 'No', 'no reason', '00:00:00', 'Hong Kong', 'Aberdin Dais', 'Fulfilled'),

(2, 2, 'inland', 2, '08:15:10', '09:30:10', '01:15:00', '2017-01-28', 'Yes', 'weather', '00:34:45', 'Aberdin Dais', 'Berlin Bradenbourg', 'In process'),

(3, 3, 'international', 3, '09:20:14', '13:52:26', '04:32:12', '2017-01-15', 'No', 'no reason', '00:00:00', 'Bordo Merenyak', 'Algera Phitilia', 'Fulfilled'),

(4, 4, 'special', 4, '12:34:40', '14:24:00', '02:50:40', '2016-12-25', 'Yes', 'technical malfunction', '00:24:25', 'Berlin Bradenbourg', 'Luxembourg Phindel', 'Fulfilled'),

(5, 5, 'charter', 5, '14:56:59', '19:56:59', '05:00:00', '2017-02-10', 'Yes', 'weather', '00:43:16', 'Algera Phitilia', 'Bryansk', 'Fulfilled'),

(6, 6, 'goods', 6, '06:22:12', '16:56:33', '10:34:21', '2017-02-05', 'No', 'no reason', '00:00:00', 'Vilnews', 'Hong Kong', 'In process'),

(7, 7, 'international', 7, '15:10:02', '19:50:58', '04:40:56', '2017-03-28', 'No', 'no reason', '00:00:00', 'Luxembourg Phindel', 'Vilnews', 'Fulfilled'),

(8, 8, 'international', 8, '17:15:32', '01:50:59', '08:35:27', '2017-02-15', 'Yes', 'technical malfunction', '00:17:45', 'Amsterdam Spiphol', 'Dnipropetrovsk', 'Canceled'),

(9, 9, 'international', 9, '19:40:20', '22:57:40', '03:17:20', '2017-01-19', 'No', 'no reason', '00:00:00', 'Bryansk', 'Berlin Bradenbourg', 'Fulfilled'),

(10, 10, 'special', 10, '21:34:47', '02:34:47', '05:00:00', '2017-03-28', 'No', 'no reason', '00:00:00', 'Dnipropetrovsk', 'Aberdin Dais', 'Canceled');

Подальше заповнення таблиць тестовими даними реалізується за подібним принципом.

Далі всі тестові запити виконуємо у вигляді простих запитів.

* + 1. Запит1

Отримати перелік і загальне число всіх працівників аеропорту, начальників відділів, працівників вказаного відділу, за стажем роботи в аеропорту, статевою ознакою, віком, ознакою наявності та кількості дітей, за розміром заробітної плати.

select surnname, name, patronymic from staffs;

select count(\*) fROM staffs;

select \* from staffs where position IN ('Crew commander', 'Airborne mechanic', 'Security crew commander');

select \* from staffs where profession = 'Security';

select surnname, name, patronymic, work\_experience from staffs;

select \* from staffs where (gender\_ = 'Female');

select \* from staffs where (age <= 30);

select surnname, name, patronymic, childhoods, number\_of\_childs FROM staffs WHERE (number\_of\_childs >= 1);

select surnname, name, patronymic, salary\_in\_dollars from staffs where (salary\_in\_dollars <= 12000);



* + - 1. Результат запиту 1



* + - 1. Результат запиту 1
    1. Запит 2

Отримати перелік і загальне число працівників в у бригаді, по всіх відділах, у зазначеному відділі, обслуговуючих конкретний рейс, за віком, сумарної(середньої) заробітної плати у бригаді.

SELECT surnname, name, patronymic, №\_service\_brigade FROM staffs WHERE (№\_service\_brigade = 722);

SELECT count(\*) FROM staffs WHERE (№\_service\_brigade = 722);

SELECT surnname, name, patronymic, №\_service\_brigade, profession FROM staffs where profession IN ('Navigator', 'Stewardess', 'First Class Pilot', 'Mechanic', 'Second Class Pilot', 'Secutiry');

SELECT count(\*) FROM staffs WHERE profession IN ('Navigator', 'Stewardess', 'First Class Pilot', 'Mechanic', 'Second Class Pilot', 'Secutiry');

SELECT surnname, name, patronymic, №\_service\_brigade, profession FROM staffs where profession IN ('Mechanic');

SELECT count(\*) FROM staffs WHERE profession IN ('Mechanic');

SELECT surnname, name, patronymic, №\_service\_brigade, code\_of\_flight FROM staffs WHERE (code\_of\_flight = 8);

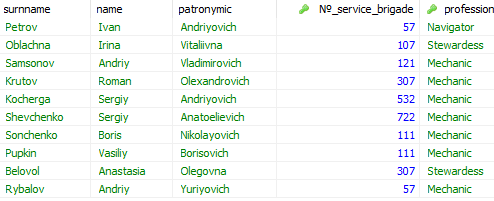
SELECT count(\*) FROM staffs WHERE (code\_of\_flight = 8);

SELECT surnname, name, patronymic, age from staffs where (age <= 35);

SELECT AVG(salary\_in\_dollars) FROM staffs WHERE (№\_service\_brigade = 722);



* + - 1. Результат запиту 2



* + - 1. Результат запиту 2
    1. Запит 3

Отримати перелік і загальне число пілотів, які пройшли медогляд або не пройшли його у вказаний рік, за статевою ознакою, віком, розміру заробітної плати.

SELECT surnname, name, patronymic, profession, body\_check, gender\_, age, salary\_in\_dollars FROM staffs WHERE (body\_check) BETWEEN '2016-01-01' AND '2016-12-31' AND gender\_ = 'Male' AND age <=40 AND salary\_in\_dollars <= 23000 AND profession = 'Pilot';

SELECT count(\*) FROM staffs WHERE (body\_check) BETWEEN '2016-01-01' AND '2016-12-31' AND gender\_ = 'Male' AND age <=40 AND salary\_in\_dollars <= 23000 AND profession = 'Pilot';



* + - 1. Результат запиту 3
    1. Запит 4

Отримати перелік і загальне число літаків приписаних до аеропорту, що знаходяться в ньому в зазначений час, за часом надходження в аеропорт, за кількістю скоєних рейсів.

SELECT plane\_model, airport\_arrival\_time, airport\_of\_origin number\_of\_flights FROM planes WHERE airport\_arrival\_time = '2016-10-02'

AND airport\_of\_origin = 'Aberdin Dais' AND number\_of\_flights <=35;

SELECT count(\*) FROM planes WHERE airport\_arrival\_time = '2016-10-02'

AND airport\_of\_origin = 'Aberdin Dais' AND number\_of\_flights <=35;



* + - 1. Результат запиту 4
    1. Запит 5

Отримати перелік і загальне число літаків, що пройшли техогляд за певний момент часу, відправлених в ремонт в зазначений час, ремонтувалися задане число разів, за кількістю скоєних рейсів до ремонту, за віком літака.

SELECT plane\_model, date\_of\_technical\_inspaction, date\_of\_the\_last\_repair, number\_of\_repairs,

number\_of\_flights\_before\_repair, year\_of\_operation

FROM planes WHERE (date\_of\_technical\_inspaction) BETWEEN '2017-01-01' AND '2017-02-26'

AND date\_of\_the\_last\_repair = '2017-02-22' AND number\_of\_repairs = 1 AND number\_of\_flights\_before\_repair <= 20

AND year\_of\_operation <= 2;

SELECT count(\*) FROM planes WHERE (date\_of\_technical\_inspaction) BETWEEN '2017-01-01' AND '2017-02-26'

AND date\_of\_the\_last\_repair = '2017-02-22' AND number\_of\_repairs = 1 AND number\_of\_flights\_before\_repair <= 20

AND year\_of\_operation <= 2;



* + - 1. Результат запиту 5
    1. Запит 6

Отримати перелік і загальне число рейсів, за вказаним маршрутом, по тривалості перельоту, за ціною квитка і за всіма цими критеріями відразу.

SELECT code\_flight, send\_airport, coming\_airport, time\_of\_flighting, code\_of\_ticket, cost FROM flights, log\_of\_tickets WHERE send\_airport = 'Vilnews' AND coming\_airport = 'Hong Kong' AND time\_of\_flighting = '10:34:21' AND cost = 991;

SELECT count(\*) FROM flights, log\_of\_tickets WHERE send\_airport = 'Vilnews' AND coming\_airport = 'Hong Kong' AND time\_of\_flighting = '10:34:21' AND cost = 991;



* + - 1. Результат запиту 6
    1. Запит 7

Отримати перелік і загальне число скасованих рейсів повністю, у зазначеному напрямку, за вказаним маршрутом, за кількістю незатребуваних місць, за відсотковим співвідношенням незатребуваних місць.

SELECT \* FROM flights WHERE status\_\_ = 'Canceled';

SELECT count(\*) FROM flights WHERE status\_\_ = 'Canceled';

SELECT \* FROM flights WHERE coming\_airport = 'Aberdin Dais';

SELECT count(\*) FROM flights WHERE coming\_airport = 'Aberdin Dais';

SELECT \* FROM flights WHERE send\_airport = 'Algera Phitilia' AND coming\_airport = 'Bryansk';

SELECT count(\*) FROM flights WHERE send\_airport = 'Algera Phitilia' AND coming\_airport = 'Bryansk';

SELECT quantity\_of\_passg\_places, number\_of\_saled, quantity\_of\_passg\_places - number\_of\_saled AS number\_of\_not\_purchesed

FROM planes\_models, log\_of\_tickets WHERE status\_ = 'Returned' AND code\_log\_ticket = 8;

SELECT quantity\_of\_passg\_places, number\_of\_saled, 100 \* number\_of\_saled / SUM(number\_of\_saled) AS percent

FROM planes\_models, log\_of\_tickets;



* + - 1. Результат запиту 7



* + - 1. Результат запиту 7
    1. Запит 8

Отримати перелік і загальне число затриманих рейсів повністю, із зазначеної причини, за вказаним маршрутом, і кількість зданих квитків за час затримки.

SELECT \* FROM flights WHERE delay = 'Yes';

SELECT count(\*) FROM flights WHERE delay = 'Yes';

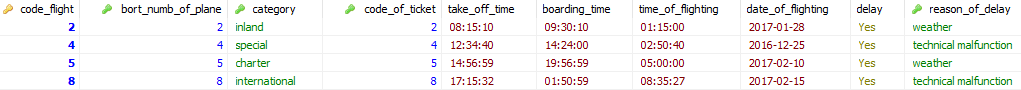
SELECT \* FROM flights WHERE delay = 'Yes' AND reason\_of\_delay = 'weather';

SELECT count(\*) FROM flights WHERE delay = 'Yes' AND reason\_of\_delay = 'weather';

SELECT \* FROM flights WHERE delay = 'Yes' AND send\_airport = 'Berlin Bradenbourg' AND coming\_airport = 'Luxembourg Phindel';

SELECT count(\*) FROM flights WHERE delay = 'Yes' AND send\_airport = 'Berlin Bradenbourg' AND coming\_airport = 'Luxembourg Phindel';

SELECT count(\*) FROM flights, log\_of\_tickets WHERE delay = 'Yes' AND status\_ = 'Returned';



* + - 1. Результат запиту 8



* + - 1. Результат запиту 8
    1. Запит 9

Отримати перелік і загальне число рейсів, якими літають літаки заданого типу, і середня кількість проданих квитків на певні маршрути, по тривалості перельоту, за ціною квитка, часу вильоту.

SELECT \* FROM flights, planes\_models WHERE bort\_numb\_of\_plane = 3 AND name\_of\_model = 'Boing-737';

SELECT count(\*) FROM flights, planes\_models WHERE bort\_numb\_of\_plane = 3 AND name\_of\_model = 'Boing-737';

SELECT AVG(number\_of\_saled) FROM log\_of\_tickets, flights WHERE send\_airport = 'Hong Kong'

AND coming\_airport = 'Aberdin Dais' AND time\_of\_flighting <= "04:00:00" AND cost <= 450

AND (date\_of\_flighting) BETWEEN '2016-08-01' AND '2016-09-30';

SELECT AVG(number\_of\_saled) FROM log\_of\_tickets, flights WHERE send\_airport = 'Luxembourg Phindel'

AND coming\_airport = 'Vilnews' AND time\_of\_flighting <= "05:00:00"

AND cost <= 800 AND (date\_of\_flighting) BETWEEN '2017-03-01' AND '2017-03-31';



* + - 1. Результат запиту 9



* + - 1. Результат запиту 9
    1. Запит 10

Отримати перелік і загальне число авіарейсів зазначеної категорії, в певному напрямку, з вказаним типом літака.

SELECT \* FROM flights, planes\_models WHERE category = 'charter' AND coming\_airport = 'Bryansk'

AND name\_of\_model = 'Boing-747';

SELECT count(\*) FROM flights, planes\_models WHERE category = 'charter' AND coming\_airport = 'Bryansk'

AND name\_of\_model = 'Boing-747';



* + - 1. Результат запиту 10
    1. Запит 11

Отримати перелік і загальне число пасажирів на даному рейсі, які відлетіли у зазначений день, відлетіли за кордон у зазначений день, за ознакою здачі речей в багажне відділення, за статевою ознакою, за віком.

SELECT \* FROM passegers, flights WHERE date\_of\_flighting = '2017-01-28' AND №\_of\_flight = 2;

SELECT count(\*) FROM passegers, flights WHERE date\_of\_flighting = '2017-01-28' AND №\_of\_flight = 2;

SELECT \* FROM passegers, flights WHERE date\_of\_flighting = '2017-03-28' AND №\_of\_flight = 7 AND category = 'international' AND gender = 'Male';

SELECT count(\*) FROM passegers, flights WHERE date\_of\_flighting = '2017-03-28' AND №\_of\_flight = 7 AND category = 'international' AND gender = 'Male';

SELECT \* FROM passegers, flights, the\_baggage WHERE code\_flight = 4 AND code\_bagg = 4 AND №\_of\_flight = 4;

SELECT count(\*) FROM passegers, flights, the\_baggage WHERE code\_flight = 4 AND code\_bagg = 4 AND №\_of\_flight = 4;

SELECT \* FROM passegers, flights WHERE code\_flight = 9 AND №\_of\_flight = 9 AND age <= 25;

SELECT count(\*) FROM passegers, flights WHERE code\_flight = 9 AND №\_of\_flight = 9 AND age <= 25;



* + - 1. Результат запиту 11
    1. Запит 12

Отримати перелік і загальне число вільних і заброньованих місць на зазначеному рейсі, на визначений день, за вказаним маршрутом, за ціною, за часом вильоту.

SELECT quantity\_of\_passg\_places, number\_of\_saled, quantity\_of\_passg\_places - number\_of\_saled AS number\_of\_free\_places

FROM planes\_models, log\_of\_tickets, flights WHERE code\_model = 2 AND code\_log\_ticket = 2 AND date\_of\_flighting = '2017-01-28'

AND send\_airport = 'Aberdin Dais' AND coming\_airport = 'Berlin Bradenbourg' AND cost <=600 AND take\_off\_time = "08:15:10";

SELECT \* FROM log\_of\_tickets, flights WHERE status\_ = 'Booked' AND code\_flight = 2 AND code\_log\_ticket = 2;

SELECT count(\*) FROM log\_of\_tickets, flights WHERE status\_ = 'Booked' AND code\_flight = 2 AND code\_log\_ticket = 2;



* + - 1. Результат запиту 12

Дамп бази даних знаходиться в Додатку В.

Висновки

База даних – це сукупність структурованих і взаємозалежних даних і методів, що забезпечують додавання, вибірку і відображення даних.

Розроблена в даному курсовому проекті база даних відповідає реляційній моделі предметного середовища в межах поставленого навчального завдання, містить набір зв’язаних між собою таблиць, наповнених наборами тестових даних, що створені у відповідності до логіки завдання.

Базу даних було реалізовано в середовищі СУБД MySQL 5.5 з використанням графічних клієнтів СУБД – HeidiSQL 9.4 portable та MySQLWorkbench 6.3. Такий вибір був обумовлений поставленим навчальним завданням.

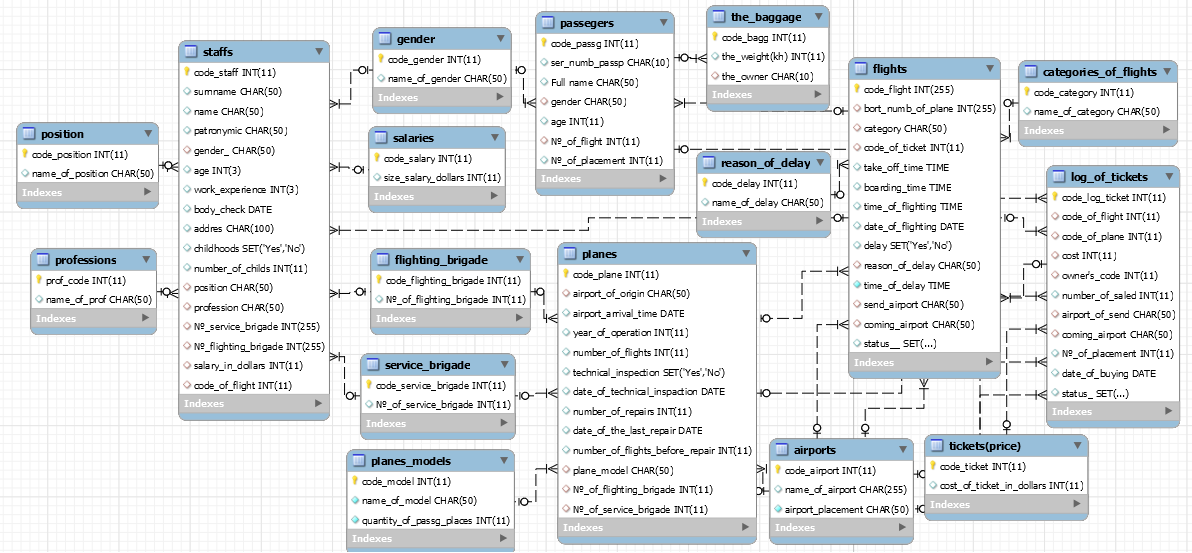
До БД було створено запити мовою SQL, що дозволили вдосконалити в ході роботи над проектом структуру БД та переконатись у працездатності проекту та вірності прийнятих в ході проектування рішень.

Під час роботи над курсовим проектом були поглиблені та закріплені знання з мови SQL та створення SQL-запитів.

Створену в ході проектування БД можна використовувати на практиці.

Перелік Використаних джерел

1. Рудяков, А.В. Методичні рекомендації щодо виконання ІНДЗ з предмету «Бази даних» // А.В. Рудяков [Текст] – Жовті Води : ОКВНЗ «ІП «Стратегія», 2015. – С. 13
2. Андрейшина Н.Б., Міхальчук В.І., Харін С.А., Рудяков А.В. Методичні вказівки до оформлення текстових документів для студентів денної форми навчання за напрямом підготовки 6.050103 «Програмна інженерія» освітньо-кваліфікаційного рівня «бакалавр» та спеціальністю 121 «Інженерія програмного забезпечення» освітньо-кваліфікаційного рівня «спеціаліст». – Жовті Води : КВНЗ ІП «Стратегія» ДОР», 2016. – 37 с.
3. Кодд, Э.Ф. Реляционная модель данных для больших совместно используемых банков данных. – СУБД, № 1, 1995 - [Електронний ресурс]. – Режим доступу : http://www.osp.ru/dbms/1995/01/01.htm. – Назва з екрану.
4. Чаповська, Р. ; Жмуркевич, А. Робота з базами даних Microsoft Access 2000. Навчальний посібник. – К.: ЦУЛ, 2003. – С.324
5. Грошев, А.С. Базы данных: Учебное пособие // А.С. Грошев [Текст] – Архангельск: Изд-во Арханг. гос. техн. ун-та, 2005. – С. 124.
6. Чен, П.П. Модель «сущность-связь» - шаг к единому представлению данных. - СУБД, №3, 1995 - [Електронний ресурс]. – Режим доступу: http://www.osp.ru/dbms/1995/03/271.html – Назва з екрану.
7. Кузнецов, С.Д. Основы современных баз данных. Лекция 14. Стандартный язык баз данных SQL. - [Електронний ресурс]. – Режим доступу: http://www.citforum.ru/database/osbd/. – Назва з екрану.
   * + - 1.   
            er-діаграма бд
         2. діаграма зв’язків між таблицями бд



* + - * 1. Дамп бази даних

-- --------------------------------------------------------

-- Хост: 127.0.0.1

-- Версия сервера: 5.5.23 - MySQL Community Server (GPL)

-- Операционная система: Win32

-- HeidiSQL Версия: 9.4.0.5125

-- --------------------------------------------------------

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!50503 SET NAMES utf8mb4 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

-- Дамп структуры базы данных aeroport

CREATE DATABASE IF NOT EXISTS `aeroport` /\*!40100 DEFAULT CHARACTER SET utf8 \*/;

USE `aeroport`;

-- Дамп структуры для таблица aeroport.airports

CREATE TABLE IF NOT EXISTS `airports` (

`code\_airport` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_airport` char(255) DEFAULT NULL,

`airport\_placement` char(50) NOT NULL DEFAULT '0',

PRIMARY KEY (`code\_airport`),

KEY `name\_of\_airport` (`name\_of\_airport`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.airports: ~11 rows (приблизительно)

/\*!40000 ALTER TABLE `airports` DISABLE KEYS \*/;

INSERT INTO `airports` (`code\_airport`, `name\_of\_airport`, `airport\_placement`) VALUES

(1, 'Algera Phitilia', 'Italy'),

(2, 'Amsterdam Spiphol', 'Netherlands'),

(3, 'Bryansk', 'Russia'),

(4, 'Vilnews', 'Lithuania'),

(5, 'Bordo Merenyak', 'France'),

(6, 'Luxembourg Phindel', 'Luxembourg'),

(7, 'Aberdin Dais', 'England'),

(8, 'Berlin Bradenbourg', 'Germany'),

(9, 'Hong Kong', 'China'),

(10, 'Dnipropetrovsk', 'Ukraine'),

(11, 'Wece', 'Germany');

/\*!40000 ALTER TABLE `airports` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.categories\_of\_flights

CREATE TABLE IF NOT EXISTS `categories\_of\_flights` (

`code\_category` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_category` char(50) DEFAULT NULL,

PRIMARY KEY (`code\_category`),

KEY `name\_of\_category` (`name\_of\_category`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.categories\_of\_flights: ~5 rows (приблизительно)

/\*!40000 ALTER TABLE `categories\_of\_flights` DISABLE KEYS \*/;

INSERT INTO `categories\_of\_flights` (`code\_category`, `name\_of\_category`) VALUES

(3, 'charter'),

(4, 'goods'),

(1, 'inland'),

(2, 'international'),

(5, 'special');

/\*!40000 ALTER TABLE `categories\_of\_flights` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.flighting\_brigade

CREATE TABLE IF NOT EXISTS `flighting\_brigade` (

`code\_flighting\_brigade` int(11) NOT NULL AUTO\_INCREMENT,

`№\_of\_flighting\_brigade` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_flighting\_brigade`),

KEY `№\_of\_flighting\_brigade` (`№\_of\_flighting\_brigade`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.flighting\_brigade: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `flighting\_brigade` DISABLE KEYS \*/;

INSERT INTO `flighting\_brigade` (`code\_flighting\_brigade`, `№\_of\_flighting\_brigade`) VALUES

(4, 232),

(5, 253),

(7, 335),

(9, 357),

(1, 567),

(6, 654),

(2, 753),

(8, 754),

(10, 789),

(3, 994);

/\*!40000 ALTER TABLE `flighting\_brigade` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.flights

CREATE TABLE IF NOT EXISTS `flights` (

`code\_flight` int(255) NOT NULL AUTO\_INCREMENT,

`bort\_numb\_of\_plane` int(255) DEFAULT NULL,

`category` char(50) DEFAULT NULL,

`code\_of\_ticket` int(11) DEFAULT NULL,

`take\_off\_time` time DEFAULT '00:00:00',

`boarding\_time` time DEFAULT '00:00:00',

`time\_of\_flighting` time DEFAULT '00:00:00',

`date\_of\_flighting` date DEFAULT NULL,

`delay` set('Yes','No') DEFAULT NULL,

`reason\_of\_delay` char(50) DEFAULT NULL,

`time\_of\_delay` time NOT NULL DEFAULT '00:00:00',

`send\_airport` char(50) DEFAULT NULL,

`coming\_airport` char(50) DEFAULT NULL,

`status\_\_` set('Fulfilled','In process','Canceled') DEFAULT NULL,

PRIMARY KEY (`code\_flight`),

KEY `FK\_flights\_airport` (`send\_airport`),

KEY `FK\_flights\_airport\_2` (`coming\_airport`),

KEY `FK\_flights\_planes` (`bort\_numb\_of\_plane`),

KEY `FK\_flights\_categories\_of\_flights` (`category`),

KEY `FK\_flights\_reason\_of\_delay` (`reason\_of\_delay`),

KEY `code\_of\_ticket` (`code\_of\_ticket`),

CONSTRAINT `FK\_flights\_reason\_of\_delay` FOREIGN KEY (`reason\_of\_delay`) REFERENCES `reason\_of\_delay` (`name\_of\_delay`),

CONSTRAINT `FK\_flights\_airport` FOREIGN KEY (`send\_airport`) REFERENCES `airports` (`name\_of\_airport`),

CONSTRAINT `FK\_flights\_airport\_2` FOREIGN KEY (`coming\_airport`) REFERENCES `airports` (`name\_of\_airport`),

CONSTRAINT `FK\_flights\_categories\_of\_flights` FOREIGN KEY (`category`) REFERENCES `categories\_of\_flights` (`name\_of\_category`),

CONSTRAINT `FK\_flights\_log\_of\_tickets` FOREIGN KEY (`code\_of\_ticket`) REFERENCES `log\_of\_tickets` (`code\_log\_ticket`),

CONSTRAINT `FK\_flights\_planes` FOREIGN KEY (`bort\_numb\_of\_plane`) REFERENCES `planes` (`code\_plane`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.flights: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `flights` DISABLE KEYS \*/;

INSERT INTO `flights` (`code\_flight`, `bort\_numb\_of\_plane`, `category`, `code\_of\_ticket`, `take\_off\_time`, `boarding\_time`, `time\_of\_flighting`, `date\_of\_flighting`, `delay`, `reason\_of\_delay`, `time\_of\_delay`, `send\_airport`, `coming\_airport`, `status\_\_`) VALUES

(1, 1, 'international', 1, '10:24:32', '13:44:52', '03:20:20', '2016-09-10', 'No', 'no reason', '00:00:00', 'Hong Kong', 'Aberdin Dais', 'Fulfilled'),

(2, 2, 'inland', 2, '08:15:10', '09:30:10', '01:15:00', '2017-01-28', 'Yes', 'weather', '00:34:45', 'Aberdin Dais', 'Berlin Bradenbourg', 'In process'),

(3, 3, 'international', 3, '09:20:14', '13:52:26', '04:32:12', '2017-01-15', 'No', 'no reason', '00:00:00', 'Bordo Merenyak', 'Algera Phitilia', 'Fulfilled'),

(4, 4, 'special', 4, '12:34:40', '14:24:00', '02:50:40', '2016-12-25', 'Yes', 'technical malfunction', '00:24:25', 'Berlin Bradenbourg', 'Luxembourg Phindel', 'Fulfilled'),

(5, 5, 'charter', 5, '14:56:59', '19:56:59', '05:00:00', '2017-02-10', 'Yes', 'weather', '00:43:16', 'Algera Phitilia', 'Bryansk', 'Fulfilled'),

(6, 6, 'goods', 6, '06:22:12', '16:56:33', '10:34:21', '2017-02-05', 'No', 'no reason', '00:00:00', 'Vilnews', 'Hong Kong', 'In process'),

(7, 7, 'international', 7, '15:10:02', '19:50:58', '04:40:56', '2017-03-28', 'No', 'no reason', '00:00:00', 'Luxembourg Phindel', 'Vilnews', 'Fulfilled'),

(8, 8, 'international', 8, '17:15:32', '01:50:59', '08:35:27', '2017-02-15', 'Yes', 'technical malfunction', '00:17:45', 'Amsterdam Spiphol', 'Dnipropetrovsk', 'Canceled'),

(9, 9, 'international', 9, '19:40:20', '22:57:40', '03:17:20', '2017-01-19', 'No', 'no reason', '00:00:00', 'Bryansk', 'Berlin Bradenbourg', 'Fulfilled'),

(10, 10, 'special', 10, '21:34:47', '02:34:47', '05:00:00', '2017-03-28', 'No', 'no reason', '00:00:00', 'Dnipropetrovsk', 'Aberdin Dais', 'Canceled');

/\*!40000 ALTER TABLE `flights` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.gender

CREATE TABLE IF NOT EXISTS `gender` (

`code\_gender` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_gender` char(50) DEFAULT NULL,

PRIMARY KEY (`code\_gender`),

KEY `name\_of\_gender` (`name\_of\_gender`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8 COMMENT='pol';

-- Дамп данных таблицы aeroport.gender: ~2 rows (приблизительно)

/\*!40000 ALTER TABLE `gender` DISABLE KEYS \*/;

INSERT INTO `gender` (`code\_gender`, `name\_of\_gender`) VALUES

(2, 'Female'),

(1, 'Male');

/\*!40000 ALTER TABLE `gender` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.log\_of\_tickets

CREATE TABLE IF NOT EXISTS `log\_of\_tickets` (

`code\_log\_ticket` int(11) NOT NULL AUTO\_INCREMENT,

`code\_of\_flight` int(11) DEFAULT NULL,

`code\_of\_plane` int(11) DEFAULT NULL,

`cost` int(11) DEFAULT NULL,

`owner's\_code` int(11) DEFAULT NULL,

`number\_of\_saled` int(11) DEFAULT NULL,

`airport\_of\_send` char(50) DEFAULT NULL,

`coming\_\_airport` char(50) DEFAULT NULL,

`№\_of\_placement` int(11) DEFAULT NULL,

`date\_of\_buying` date DEFAULT NULL,

`status\_` set('Purchased','Booked','Returned') DEFAULT NULL,

PRIMARY KEY (`code\_log\_ticket`),

KEY `FK\_log of tickets\_flights` (`code\_of\_flight`),

KEY `FK\_log of tickets\_passegers` (`owner's\_code`),

KEY `FK\_log\_of\_tickets\_airport` (`airport\_of\_send`),

KEY `FK\_log\_of\_tickets\_planes` (`code\_of\_plane`),

KEY `FK\_log\_of\_tickets\_price\_of\_tickets` (`cost`),

KEY `FK\_log\_of\_tickets\_airports` (`coming\_\_airport`),

CONSTRAINT `FK\_log\_of\_tickets\_airports` FOREIGN KEY (`coming\_\_airport`) REFERENCES `airports` (`name\_of\_airport`),

CONSTRAINT `FK\_log of tickets\_flights` FOREIGN KEY (`code\_of\_flight`) REFERENCES `flights` (`code\_flight`),

CONSTRAINT `FK\_log of tickets\_passegers` FOREIGN KEY (`owner's\_code`) REFERENCES `passegers` (`code\_passg`),

CONSTRAINT `FK\_log\_of\_tickets\_airport` FOREIGN KEY (`airport\_of\_send`) REFERENCES `airports` (`name\_of\_airport`),

CONSTRAINT `FK\_log\_of\_tickets\_planes` FOREIGN KEY (`code\_of\_plane`) REFERENCES `planes` (`code\_plane`),

CONSTRAINT `FK\_log\_of\_tickets\_price\_of\_tickets` FOREIGN KEY (`cost`) REFERENCES `tickets(price)` (`cost\_of\_ticket\_in\_dollars`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.log\_of\_tickets: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `log\_of\_tickets` DISABLE KEYS \*/;

INSERT INTO `log\_of\_tickets` (`code\_log\_ticket`, `code\_of\_flight`, `code\_of\_plane`, `cost`, `owner's\_code`, `number\_of\_saled`, `airport\_of\_send`, `coming\_\_airport`, `№\_of\_placement`, `date\_of\_buying`, `status\_`) VALUES

(1, 1, 1, 449, 1, 1, 'Hong Kong', 'Aberdin Dais', 44, '2016-09-05', 'Purchased'),

(2, 2, 2, 600, 2, 1, 'Aberdin Dais', 'Berlin Bradenbourg', 156, '2017-01-11', 'Booked'),

(3, 3, 3, 500, 3, 1, 'Bordo Merenyak', 'Algera Phitilia', 55, '2017-01-12', 'Purchased'),

(4, 4, 4, 789, 4, 1, 'Berlin Bradenbourg', 'Luxembourg Phindel', 56, '2016-12-20', 'Purchased'),

(5, 5, 5, 1200, 5, 1, 'Algera Phitilia', 'Bryansk', 23, '2017-02-02', 'Purchased'),

(6, 6, 6, 991, 6, 1, 'Vilnews', 'Hong Kong', 12, '2017-02-01', 'Purchased'),

(7, 7, 7, 789, 7, 1, 'Luxembourg Phindel', 'Vilnews', 99, '2017-03-20', 'Booked'),

(8, 8, 8, 949, 8, 1, 'Amsterdam Spiphol', 'Dnipropetrovsk', 44, '2017-02-10', 'Returned'),

(9, 9, 9, 699, 9, 1, 'Bryansk', 'Berlin Bradenbourg', 77, '2017-01-11', 'Booked'),

(10, 10, 10, 1000, 10, 1, 'Dnipropetrovsk', 'Aberdin Dais', 123, '2017-03-24', 'Returned');

/\*!40000 ALTER TABLE `log\_of\_tickets` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.passegers

CREATE TABLE IF NOT EXISTS `passegers` (

`code\_passg` int(11) NOT NULL AUTO\_INCREMENT,

`ser\_numb\_passp` char(10) DEFAULT 'NOT NULL',

`Full name` char(50) DEFAULT 'NOT NULL',

`gender` char(50) DEFAULT NULL,

`age` int(11) DEFAULT NULL,

`№\_of\_flight` int(11) DEFAULT NULL,

`№\_of\_placement` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_passg`),

KEY `№\_of\_flight` (`№\_of\_flight`),

KEY `ser\_numb\_passp` (`ser\_numb\_passp`),

KEY `FK\_passegers\_gender` (`gender`),

CONSTRAINT `FK\_passegers\_gender` FOREIGN KEY (`gender`) REFERENCES `gender` (`name\_of\_gender`),

CONSTRAINT `FK\_\_flights` FOREIGN KEY (`№\_of\_flight`) REFERENCES `flights` (`code\_flight`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.passegers: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `passegers` DISABLE KEYS \*/;

INSERT INTO `passegers` (`code\_passg`, `ser\_numb\_passp`, `Full name`, `gender`, `age`, `№\_of\_flight`, `№\_of\_placement`) VALUES

(1, 'AO 221111', 'Kamyaniy Sergey Vladimirovich', 'Male', 24, 5, 44),

(2, 'AO 234242', 'Loboda Bogdan Olexandrovich', 'Male', 34, 6, 156),

(3, 'AO 328481', 'Polyakova Yuliya Mikolaevna', 'Female', 22, 2, 55),

(4, 'AO 332244', 'Elnikova Valeria Vitaliivna', 'Female', 33, 1, 56),

(5, 'AO 345677', 'Bondarenko Dmytro Panasovich', 'Male', 45, 3, 23),

(6, 'AO 356905', 'Yakovenko Volodimir Yuriyovich', 'Male', 27, 7, 12),

(7, 'AO 423842', 'Cherkaschenko Nikita Yuriyovich', 'Male', 54, 8, 99),

(8, 'AO 432824', 'Zakharchenko Yuri Anatoliyovich', 'Male', 22, 9, 44),

(9, 'AO 567890', 'Zelenskaya Anastasiya Sergiivna', 'Female', 19, 4, 77),

(10, 'AO 764583', 'Andreev Olexandr Vitalievich', 'Male', 20, 10, 123);

/\*!40000 ALTER TABLE `passegers` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.planes

CREATE TABLE IF NOT EXISTS `planes` (

`code\_plane` int(11) NOT NULL AUTO\_INCREMENT,

`airport\_of\_origin` char(50) DEFAULT NULL,

`airport\_arrival\_time` date DEFAULT NULL,

`year\_of\_operation` int(11) DEFAULT NULL,

`number\_of\_flights` int(11) DEFAULT NULL,

`technical\_inspection` set('Yes','No') DEFAULT NULL,

`date\_of\_technical\_inspaction` date DEFAULT NULL,

`number\_of\_repairs` int(11) DEFAULT NULL,

`date\_of\_the\_last\_repair` date DEFAULT NULL,

`number\_of\_flights\_before\_repair` int(11) DEFAULT NULL,

`plane\_model` char(50) DEFAULT NULL,

`№\_of\_flighting\_brigade` int(11) DEFAULT NULL,

`№\_of\_service\_brigade` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_plane`),

KEY `FK\_planes\_airport` (`airport\_of\_origin`),

KEY `FK\_planes\_planes\_models` (`plane\_model`),

KEY `FK\_planes\_flighting\_brigade` (`№\_of\_flighting\_brigade`),

KEY `FK\_planes\_service\_brigade` (`№\_of\_service\_brigade`),

CONSTRAINT `FK\_planes\_airport` FOREIGN KEY (`airport\_of\_origin`) REFERENCES `airports` (`name\_of\_airport`),

CONSTRAINT `FK\_planes\_flighting\_brigade` FOREIGN KEY (`№\_of\_flighting\_brigade`) REFERENCES `flighting\_brigade` (`№\_of\_flighting\_brigade`),

CONSTRAINT `FK\_planes\_planes\_models` FOREIGN KEY (`plane\_model`) REFERENCES `planes\_models` (`name\_of\_model`),

CONSTRAINT `FK\_planes\_service\_brigade` FOREIGN KEY (`№\_of\_service\_brigade`) REFERENCES `service\_brigade` (`№\_of\_service\_brigade`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.planes: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `planes` DISABLE KEYS \*/;

INSERT INTO `planes` (`code\_plane`, `airport\_of\_origin`, `airport\_arrival\_time`, `year\_of\_operation`, `number\_of\_flights`, `technical\_inspection`, `date\_of\_technical\_inspaction`, `number\_of\_repairs`, `date\_of\_the\_last\_repair`, `number\_of\_flights\_before\_repair`, `plane\_model`, `№\_of\_flighting\_brigade`, `№\_of\_service\_brigade`) VALUES

(1, 'Hong Kong', '2016-09-12', 2, 56, 'Yes', '2017-01-11', 1, '2017-01-10', 34, 'A-310', 232, 57),

(2, 'Aberdin Dais', '2016-10-02', 1, 34, 'Yes', '2017-02-10', 3, '2017-02-22', 22, 'Airbus A320', 253, 107),

(3, 'Bordo Merenyak', '2017-01-10', 3, 33, 'Yes', '2017-02-15', 3, '2017-02-14', 21, 'Airbus A321', 335, 111),

(4, 'Berlin Bradenbourg', '2017-03-14', 1, 27, 'Yes', '2017-01-23', 2, '2017-01-22', 19, 'Airbus A330', 357, 121),

(5, 'Algera Phitilia', '2017-05-22', 2, 40, 'Yes', '2017-02-17', 1, '2017-02-16', 20, 'Boing-737', 567, 123),

(6, 'Vilnews', '2017-03-20', 2, 30, 'Yes', '2017-02-23', 1, '2017-02-22', 15, 'Boing-747', 654, 307),

(7, 'Luxembourg Phindel', '2016-12-04', 3, 32, 'Yes', '2017-02-26', 1, '2017-02-22', 18, 'Boing-777', 753, 532),

(8, 'Amsterdam Spiphol', '2016-10-02', 1, 30, 'Yes', '2017-02-27', 2, '2017-02-26', 23, 'Sukhoy SuperJet 100', 754, 722),

(9, 'Bryansk', '2016-11-19', 3, 25, 'Yes', '2017-03-05', 3, '2017-03-04', 16, 'TU-144', 789, 745),

(10, 'Dnipropetrovsk', '2017-02-16', 4, 23, 'Yes', '2017-03-06', 1, '2017-03-05', 20, 'TU-154', 994, 993);

/\*!40000 ALTER TABLE `planes` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.planes\_models

CREATE TABLE IF NOT EXISTS `planes\_models` (

`code\_model` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_model` char(50) NOT NULL DEFAULT '0',

`quantity\_of\_passg\_places` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`code\_model`),

KEY `name\_of\_model` (`name\_of\_model`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.planes\_models: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `planes\_models` DISABLE KEYS \*/;

INSERT INTO `planes\_models` (`code\_model`, `name\_of\_model`, `quantity\_of\_passg\_places`) VALUES

(1, 'TU-154', 203),

(2, 'TU-144', 187),

(3, 'Boing-737', 158),

(4, 'Boing-777', 402),

(5, 'Boing-747', 356),

(6, 'A-310', 400),

(7, 'Airbus A330', 241),

(8, 'Airbus A321', 195),

(9, 'Airbus A320', 170),

(10, 'Sukhoy SuperJet 100', 87);

/\*!40000 ALTER TABLE `planes\_models` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.position

CREATE TABLE IF NOT EXISTS `position` (

`code\_position` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_position` char(50) DEFAULT NULL,

PRIMARY KEY (`code\_position`),

KEY `name\_of\_position` (`name\_of\_position`)

) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.position: ~11 rows (приблизительно)

/\*!40000 ALTER TABLE `position` DISABLE KEYS \*/;

INSERT INTO `position` (`code\_position`, `name\_of\_position`) VALUES

(3, 'Airborne mechanic'),

(9, 'Chief mechanical engineer'),

(1, 'Crew commander'),

(8, 'Electrician'),

(2, 'First assistant'),

(7, 'Mechanic'),

(5, 'Navigator'),

(6, 'Refueler'),

(11, 'Security crew commander'),

(10, 'Secutiry crew'),

(4, 'Stewardess');

/\*!40000 ALTER TABLE `position` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.professions

CREATE TABLE IF NOT EXISTS `professions` (

`prof\_code` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_prof` char(50) DEFAULT NULL,

PRIMARY KEY (`prof\_code`),

KEY `name\_of\_prof` (`name\_of\_prof`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.professions: ~5 rows (приблизительно)

/\*!40000 ALTER TABLE `professions` DISABLE KEYS \*/;

INSERT INTO `professions` (`prof\_code`, `name\_of\_prof`) VALUES

(5, 'Mechanic'),

(3, 'Navigator'),

(1, 'Pilot'),

(6, 'Security'),

(4, 'Stewardess');

/\*!40000 ALTER TABLE `professions` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.reason\_of\_delay

CREATE TABLE IF NOT EXISTS `reason\_of\_delay` (

`code\_delay` int(11) NOT NULL AUTO\_INCREMENT,

`name\_of\_delay` char(50) DEFAULT NULL,

PRIMARY KEY (`code\_delay`),

KEY `name\_of\_delay` (`name\_of\_delay`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.reason\_of\_delay: ~3 rows (приблизительно)

/\*!40000 ALTER TABLE `reason\_of\_delay` DISABLE KEYS \*/;

INSERT INTO `reason\_of\_delay` (`code\_delay`, `name\_of\_delay`) VALUES

(3, 'no reason'),

(2, 'technical malfunction'),

(1, 'weather');

/\*!40000 ALTER TABLE `reason\_of\_delay` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.salaries

CREATE TABLE IF NOT EXISTS `salaries` (

`code\_salary` int(11) NOT NULL AUTO\_INCREMENT,

`size\_salary\_dollars` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_salary`),

KEY `size\_salary\_dollars` (`size\_salary\_dollars`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.salaries: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `salaries` DISABLE KEYS \*/;

INSERT INTO `salaries` (`code\_salary`, `size\_salary\_dollars`) VALUES

(1, 3000),

(2, 5000),

(3, 7000),

(4, 10000),

(5, 12000),

(6, 14000),

(7, 15000),

(8, 20000),

(9, 22000),

(10, 23000);

/\*!40000 ALTER TABLE `salaries` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.service\_brigade

CREATE TABLE IF NOT EXISTS `service\_brigade` (

`code\_service\_brigade` int(11) NOT NULL AUTO\_INCREMENT,

`№\_of\_service\_brigade` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_service\_brigade`),

KEY `№\_of\_service\_brigade` (`№\_of\_service\_brigade`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.service\_brigade: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `service\_brigade` DISABLE KEYS \*/;

INSERT INTO `service\_brigade` (`code\_service\_brigade`, `№\_of\_service\_brigade`) VALUES

(6, 57),

(9, 107),

(8, 111),

(1, 121),

(3, 123),

(2, 307),

(10, 532),

(4, 722),

(7, 745),

(5, 993);

/\*!40000 ALTER TABLE `service\_brigade` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.staffs

CREATE TABLE IF NOT EXISTS `staffs` (

`code\_staff` int(11) NOT NULL AUTO\_INCREMENT,

`surnname` char(50) DEFAULT NULL,

`name` char(50) DEFAULT NULL,

`patronymic` char(50) DEFAULT NULL,

`gender\_` char(50) DEFAULT NULL,

`age` int(3) DEFAULT NULL,

`work\_experience` int(3) DEFAULT NULL,

`body\_check` date DEFAULT NULL,

`addres` char(100) DEFAULT NULL,

`childhoods` set('Yes','No') DEFAULT NULL,

`number\_of\_childs` int(11) DEFAULT NULL,

`position` char(50) DEFAULT NULL,

`profession` char(50) DEFAULT NULL,

`№\_service\_brigade` int(255) DEFAULT NULL,

`№\_flighting\_brigade` int(255) DEFAULT NULL,

`salary\_in\_dollars` int(11) DEFAULT NULL,

`code\_of\_flight` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_staff`),

KEY `FK\_\_gender` (`gender\_`),

KEY `FK\_staffs\_salaries` (`salary\_in\_dollars`),

KEY `FK\_staffs\_position` (`position`),

KEY `FK\_staffs\_professions` (`profession`),

KEY `FK\_staffs\_service\_brigade` (`№\_service\_brigade`),

KEY `FK\_staffs\_flighting\_brigade` (`№\_flighting\_brigade`),

KEY `code\_of\_flight` (`code\_of\_flight`),

CONSTRAINT `FK\_staffs\_flighting\_brigade` FOREIGN KEY (`№\_flighting\_brigade`) REFERENCES `flighting\_brigade` (`№\_of\_flighting\_brigade`),

CONSTRAINT `FK\_staffs\_flights` FOREIGN KEY (`code\_of\_flight`) REFERENCES `flights` (`code\_flight`),

CONSTRAINT `FK\_staffs\_gender` FOREIGN KEY (`gender\_`) REFERENCES `gender` (`name\_of\_gender`),

CONSTRAINT `FK\_staffs\_position` FOREIGN KEY (`position`) REFERENCES `position` (`name\_of\_position`),

CONSTRAINT `FK\_staffs\_professions` FOREIGN KEY (`profession`) REFERENCES `professions` (`name\_of\_prof`),

CONSTRAINT `FK\_staffs\_salaries` FOREIGN KEY (`salary\_in\_dollars`) REFERENCES `salaries` (`size\_salary\_dollars`),

CONSTRAINT `FK\_staffs\_service\_brigade` FOREIGN KEY (`№\_service\_brigade`) REFERENCES `service\_brigade` (`№\_of\_service\_brigade`)

) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.staffs: ~15 rows (приблизительно)

/\*!40000 ALTER TABLE `staffs` DISABLE KEYS \*/;

INSERT INTO `staffs` (`code\_staff`, `surnname`, `name`, `patronymic`, `gender\_`, `age`, `work\_experience`, `body\_check`, `addres`, `childhoods`, `number\_of\_childs`, `position`, `profession`, `№\_service\_brigade`, `№\_flighting\_brigade`, `salary\_in\_dollars`, `code\_of\_flight`) VALUES

(1, 'Petrov', 'Ivan', 'Andriyovich', 'Male', 32, 3, '2017-01-12', 'Kyiv', 'Yes', 2, 'Navigator', 'Navigator', 57, 232, 12000, 1),

(2, 'Oblachna', 'Irina', 'Vitaliivna', 'Female', 25, 1, '2017-01-11', 'Dnipro', 'No', 0, 'Stewardess', 'Stewardess', 107, 253, 3000, 2),

(3, 'Bokoviy', 'Volodimir', 'Sergiyovich', 'Male', 39, 7, '2016-12-27', 'Kharkiv', 'Yes', 1, 'Crew commander', 'Pilot', 111, 335, 23000, 3),

(4, 'Samsonov', 'Andriy', 'Vladimirovich', 'Male', 29, 2, '2016-11-21', 'Dnipro', 'Yes', 3, 'Electrician', 'Mechanic', 121, 357, 14000, 4),

(5, 'Kutkovetsky', 'Volodimir', 'Valeriovich', 'Male', 27, 2, '2017-01-10', 'Zvenigorodka', 'Yes', 3, 'First assistant', 'Pilot', 123, 567, 15000, 5),

(6, 'Krutov', 'Roman', 'Olexandrovich', 'Male', 30, 3, '2016-09-28', 'Moscow', 'No', 0, 'Refueler', 'Mechanic', 307, 654, 12000, 6),

(7, 'Kocherga', 'Sergiy', 'Andriyovich', 'Male', 34, 4, '2017-02-09', 'Dnipro', 'No', 0, 'Airborne mechanic', 'Mechanic', 532, 753, 20000, 7),

(8, 'Shevchenko', 'Sergiy', 'Anatoelievich', 'Male', 26, 1, '2016-08-05', 'Lviv', 'Yes', 1, 'Mechanic', 'Mechanic', 722, 754, 7000, 8),

(9, 'Malov', 'Vladislav', 'Vladislavovich', 'Male', 28, 2, '2017-03-01', 'Kriviy Righ', 'No', 0, 'Security crew commander', 'Security', 745, 789, 10000, 9),

(10, 'Sushkevich', 'Anatolyi', 'Sergiyovich', 'Male', 33, 4, '2017-03-27', 'Dnipro', 'Yes', 2, 'Secutiry crew', 'Security', 993, 994, 5000, 10),

(11, 'Sonchenko', 'Boris', 'Nikolayovich', 'Male', 25, 3, '2017-04-06', 'Dnipro', 'No', 0, 'Electrician', 'Mechanic', 111, 253, 14000, 3),

(12, 'Pupkin', 'Vasiliy', 'Borisovich', 'Male', 24, 1, '2017-05-11', 'Lviv', 'No', 0, 'Refueler', 'Mechanic', 111, 335, 12000, 3),

(13, 'Belovol', 'Anastasia', 'Olegovna', 'Female', 27, 2, '2017-02-12', 'Kyiv', 'Yes', 1, 'Stewardess', 'Stewardess', 307, 654, 3000, 6),

(14, 'Rybalov', 'Andriy', 'Yuriyovich', 'Male', 30, 3, '2016-12-23', 'Dnipro', 'No', 0, 'Electrician', 'Mechanic', 57, 232, 14000, 1),

(15, 'Rudenko', 'Maxim', 'Valeriovich', 'Male', 28, 2, '2017-01-09', 'Kryviy Righ', 'No', 0, 'Secutiry crew', 'Security', 722, 754, 5000, 8);

/\*!40000 ALTER TABLE `staffs` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.the\_baggage

CREATE TABLE IF NOT EXISTS `the\_baggage` (

`code\_bagg` int(11) NOT NULL AUTO\_INCREMENT,

`the\_weight(kh)` int(11) DEFAULT NULL,

`the\_owner` char(10) DEFAULT NULL,

PRIMARY KEY (`code\_bagg`),

KEY `FK\_the\_baggage\_passegers` (`the\_owner`),

CONSTRAINT `FK\_the\_baggage\_passegers` FOREIGN KEY (`the\_owner`) REFERENCES `passegers` (`ser\_numb\_passp`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.the\_baggage: ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `the\_baggage` DISABLE KEYS \*/;

INSERT INTO `the\_baggage` (`code\_bagg`, `the\_weight(kh)`, `the\_owner`) VALUES

(1, 5, 'AO 221111'),

(2, 3, 'AO 764583'),

(3, 6, 'AO 345677'),

(4, 4, 'AO 423842'),

(5, 2, 'AO 332244'),

(6, 7, 'AO 234242'),

(7, 10, 'AO 328481'),

(8, 11, 'AO 356905'),

(9, 5, 'AO 432824'),

(10, 9, 'AO 567890');

/\*!40000 ALTER TABLE `the\_baggage` ENABLE KEYS \*/;

-- Дамп структуры для таблица aeroport.tickets(price)

CREATE TABLE IF NOT EXISTS `tickets(price)` (

`code\_ticket` int(11) NOT NULL AUTO\_INCREMENT,

`cost\_of\_ticket\_in\_dollars` int(11) DEFAULT NULL,

PRIMARY KEY (`code\_ticket`),

KEY `cost\_of\_ticket\_in\_dollars` (`cost\_of\_ticket\_in\_dollars`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8;

-- Дамп данных таблицы aeroport.tickets(price): ~10 rows (приблизительно)

/\*!40000 ALTER TABLE `tickets(price)` DISABLE KEYS \*/;

INSERT INTO `tickets(price)` (`code\_ticket`, `cost\_of\_ticket\_in\_dollars`) VALUES

(1, 449),

(2, 500),

(3, 600),

(4, 699),

(5, 789),

(6, 899),

(7, 949),

(8, 991),

(9, 1000),

(10, 1200);

/\*!40000 ALTER TABLE `tickets(price)` ENABLE KEYS \*/;

/\*!40101 SET SQL\_MODE=IFNULL(@OLD\_SQL\_MODE, '') \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=IF(@OLD\_FOREIGN\_KEY\_CHECKS IS NULL, 1, @OLD\_FOREIGN\_KEY\_CHECKS) \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;