

Accelist 3+1 Training

Competency Examination

SQL PROGRAMMING EXAM 2023

Foreword

This examination is a test for evaluating the trainee understanding after learning SQL Programming Curriculum.

If there is any question about this examination, please ask the trainer.

Collect your work (**not as a zip but your source code**) by pushing a branch to origin with format **NIM-Name** (e.g 2001563082-EagarSatya) and do a merge request to the master branch.

Rules

- You may open any references, but no discussion with other participants. And of course, no cheating (copy-ing, using your friend's code). If the examiner found an indication **you will get 0 point from this examination.**
- When grading, if a **Point Killer is discovered, you will get 0 points.**
- Coding Conventions:

C# Coding Conventions:

1. **PascalCase** for every class name and class members (fields, properties, methods, etc.) names.
2. **camelCase** for every local variable (var) names and method parameter names.
3. Not await-ing an asynchronous method, which includes blocking it using: `.Result`, `.Wait()`

Accelist Coding Conventions:

1. All control structure (e.g., if-else, foreach-in) must be written using parentheses `{}`
2. Use **LINQ** (Lambda Expression / `Q => ...`) over **Query syntax** (from ...), **except when joining tables.**

Point Killer

- Melanggar aturan Coding Conventions.
- **Pastikan validasi terpasang pada Front End dan Back End pada setiap constraint.**
- Untuk fitur **Create, Update, dan View Detail** dibuat terpisah dengan **Grid** utama (**dengan route baru**). Contoh : Page Company dengan route localhost:3000/company, maka:
 - Untuk fitur **Create/Update**, berada pada route localhost:3000/company/form
 - Untuk fitur **View Detail**, nya badada pada route localhost:3000/company/detail/{id}
- Tipe data yang digunakan pada **data waktu** adalah **DateTimeOffset dengan format UTC**.
- Jika data tidak tervalidasi maka menampilkan message error dengan jelas sesuai dengan constraint yang dilanggar.

Contoh:

[Label] [Constraint yang dilanggar]

Plane Name can't be empty

Plane Name maximum character is 300

Plane Name must be unique

- Menggunakan Code First Approach pada saat pembuatan database.
- Untuk penggunaan **Offset Pagination** pastikan API:
 - Menerima query params
 - itemPerPage: int
 - currentPage: int
- Untuk penggunaan **Cursor Pagination** pastikan API mengembalikan response next/previous cursor dengan format query params yang akan di hit. Contoh:

```
{
  "cinemas": [],
  "after": "10",
  "before": "5"
}
```

- Wajib menggunakan *Model-API-Request-Validation-Response (MARVEL) Pattern*.
- Untuk return *error handling* nya wajib menggunakan format **standard RFC 7807**.
- Setiap konek ke DB harus menggunakan async await.

What to code?

Accelist International Airport ingin membuat sebuah Web Aplikasi untuk menyimpan data-data perusahaan penerbangan yang bekerja sama dengan Accelist Airport.

Buatlah sistem dengan rincian desain arsitektur berikut:

1. **Back End**

- .NET 6
- PostgreSQL
- EF Core

2. **Front End**

- Next.Js

Berikut fitur-fitur yang akan dikembangkan:

1. Setup the Environment
 - Make sure the Migrations commands are successfully executed.
2. Halaman Master Perusahaan
 - Pada front end dapat diakses melalui sidebar ataupun route **/company**
 - Terdapat fitur **Create, Read, Update, & Delete**
 - Menampilkan grid data menggunakan **Offset Pagination**.
 - Pada component Grid yang ditampilkan adalah sebagai berikut:
 - Company Name: string
 - Company Address: string
 - Company Number Phone: string
 - Logo Perusahaan: Guid
 - Gunakan Guid ini untuk memperoleh preassigned URL yang akan digunakan pada proses rendering.
 - Terdapat fitur **Delete (berupa button)** pada setiap row data grid.
 - Pada halaman **Create, Update pastikan terdapat komponen sebagai berikut:**
 - Company Name (Input type text, Required, Unique, maximum 100 character)
 - Company Address (Input type text, Required, maximum 500 character)
 - Company Number Phone (Input type text, Required, maximum 20 character)
 - Logo (Input type file, **will be upload using MinIO services that created using the schema that used on training**)
 - **Pastikan validasi terpasang pada Front End dan Back End.**
 - Pada bagian Back End API Route yang digunakan untuk halaman ini adalah **/api/v1/company**.
 - Buat sebuah GET API untuk membaca data menggunakan **Offset Pagination**
 - Buat sebuah POST API untuk membuat data perusahaan.
 - Buat sebuah POST API pada sebuah **controller terpisah** (MinIO Controller) **untuk MinIO** sebagai penunjang upload foto.
 - Buat sebuah GET API pada MinIO Controller **untuk memperoleh image logo** pada data grid.
 - Buat sebuah PUT API untuk mengupdate data perusahaan dengan validasi yang telah di state pada Front End.
 - Buat sebuah DELETE API untuk menghapus data, **dengan validasi bahwa perusahaan tersebut tidak memiliki jadwal penerbangan.**

3. Halaman Master Pesawat

- Pada front end dapat diakses melalui sidebar ataupun route **/plane**
- Terdapat fitur **Create, Read, Update, & Delete**
- Menampilkan grid data menggunakan **Offset Pagination**.
- Pada component Grid yang ditampilkan adalah sebagai berikut:
 - Plane Name: string
 - Plane Type: string
 - Plane Capacity: int
 - Plane Fuel Capacity: int
- Terdapat fitur **Delete (berupa button)** pada setiap row data grid.
- Pada halaman **Create, Update** pastikan terdapat komponen sebagai berikut:
 - Plane Name (Input type text, Required, Unique, maximum 100 character)
 - Company Name (Dropdown, Required)
 - Plane Type (Dropdown, Required)
 - Plane Capacity (Input type number, Required, minimum 1, maximum 1000)
 - Plane Fuel Capacity (Input type number, Required, minimum 1, maximum 30000)
- Pada bagian Back End API Route yang digunakan untuk halaman ini adalah **/api/v1/plane**.
- Buat sebuah GET API untuk mengambil data company pesawat.
- Buat sebuah GET API untuk mengambil data tipe pesawat.
- **Plane Type di seed pada migration (Cargo, Passenger)**.
- Buat sebuah GET API untuk membaca data menggunakan **Offset Pagination**.
- Buat sebuah POST API untuk membuat data pesawat dengan validasi yang telah di state pada Front End.
- Buat sebuah PUT API untuk mengupdate data pesawat dengan validasi yang telah di state pada Front End.
- Buat sebuah DELETE API untuk menghapus data, **dengan validasi bahwa pesawat tersebut tidak memiliki jadwal penerbangan**.

4. Halaman Transaksi Jadwal Penerbangan

- Pada front end dapat diakses melalui sidebar ataupun route **/schedule**
- Terdapat fitur **Create, Read, Update, & Delete**
- Menampilkan grid data menggunakan **Cursor Pagination**.
- Pada component Grid yang ditampilkan adalah sebagai berikut:
 - Plane Name: string
 - Company Name: string
 - Plane Capacity: int
 - Departure Date: string dengan format (dd-MMM-yyyy hh:mm)
 - Arrival Date: string dengan format (dd-MMM-yyyy hh:mm)
 - Gate Name: string
- Terdapat fitur **Delete (berupa button)** pada setiap row data grid.
- Pada halaman **Create, Update** pastikan terdapat komponen sebagai berikut:
 - Company Name (Dropdown, Not Required, only for filtering plane name)

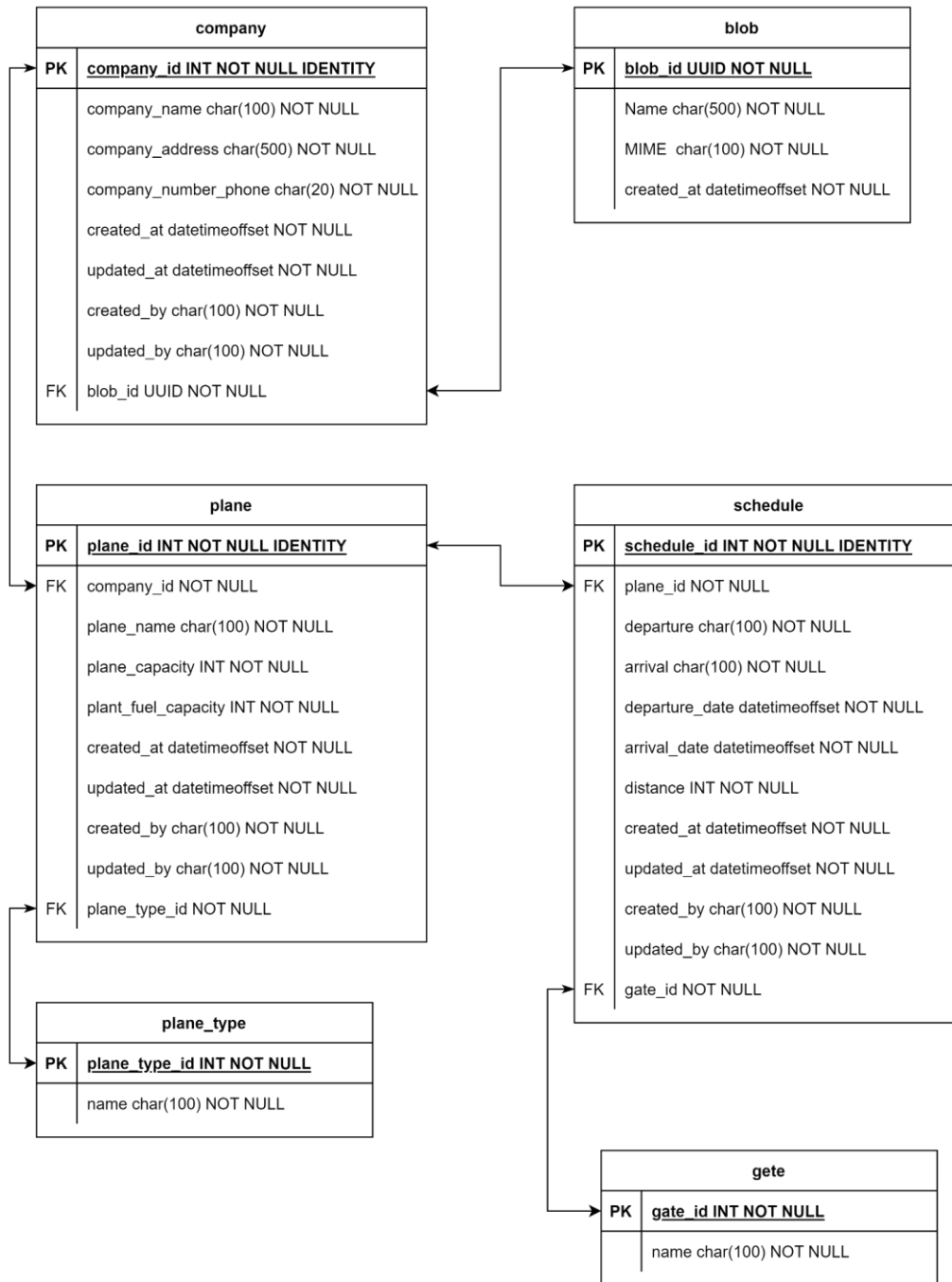
- Plane Name (Dropdown, Required)
- Gate Id (Dropdown, Required)
- Departure (Input type text, Required, maximum 100 character)
- Arrival (Input type text, Required, maximum 100 character)
- Departure Date (Input type date, Required, minimum H+7 dari hari sekarang)
- Arrival Date (Input type date, Required, minimum H+7 dari hari sekarang, dan harus lebih besar dari Departure Date).
- Distance (Input type number, Required, minimum 1, maximum 30000, akan di validasi pada back end bahwa tipe pesawat hanya mampu menempuh perjalanan dengan Distance <= Plane Fuel Capacity).
- **Pastikan validasi terpasang pada Front End dan Back End.**
- Pada bagian Back End API Route yang digunakan untuk halaman ini adalah **/api/v1/schedule**.
- Buat sebuah GET API untuk mengambil data dropdown pesawat.
- Buat sebuah GET API untuk mengambil data dropdown gate.
- **Gate Id di seed pada migration (Gate 1, Gate 2, Gate 3, Gate 4).**
- Buat sebuah GET API untuk membaca data Jadwal penerbangan menggunakan **Cursor Pagination**.
- Buat sebuah POST API untuk membuat data pesawat dengan validasi yang telah di state pada Front End.
- Buat sebuah PUT API untuk mengupdate data pesawat dengan validasi yang telah di state pada Front End.
- Buat sebuah DELETE API untuk menghapus data.

5. Halaman Dashboard

- Pada front end dapat diakses melalui sidebar ataupun route **/dashboard**
- Terdapat fitur **Read**
- Menampilkan grid data menggunakan **Offset Pagination**.
- Pada component Grid, yang ditampilkan adalah sebagai berikut:
 - Company Name: string
 - Plane Name: string
 - Total Count Penerbangan dari Pesawat Tersebut (all-time): int
- Terdapat fitur **View Detail** yang akan menampilkan grid detail yang menerima parameter Plane Id.
- Pada halaman **View Detail** yang ditampilkan adalah sebagai berikut:
 - Nama Pesawat: string
 - Jadwal penerbangan (tanggal penerbangan) dalam periode hari dalam kurun waktu 3 bulan (dari 1 bulan yang lalu hingga 1 bulan ke depan): string dengan format (dd-MMM-yyyy hh:mm)
 - **Total count penerbangan pesawat pada hari** tersebut menggunakan **Cursor Pagination: int**.
- Pada bagian Back End API Route yang digunakan untuk halaman ini adalah **/api/v1/dashboard**.
- Buat sebuah GET API untuk mengambil data main grid menggunakan **Offset Pagination**.

- Buat sebuah GET API untuk membaca data **View Detail** dengan menggunakan **Cursor Pagination**.

ERD



Preview

1. Master Perusahaan
 - a. Main Grid

accelist¹

HomeFeaturesPricingFAQAbout

Signed in as Eagar ▾

Dashboard
















Company

Plane

Flight Schedule

Company

Insert

Company Name	Company Address	Company Number Phone	Logo	Action
Company A	Palmerah	081213213123123		 
Company A	Palmerah	081213213123123		 
Company A	Palmerah	081213213123123		 
Company A	Palmerah	081213213123123		 
Company A	Palmerah	081213213123123		 

FirstLast

- b. Form

accelist¹

HomeFeaturesPricingFAQAbout

Signed in as Eagar ▾

Dashboard

Company

Plane

Flight Schedule

Form Data

Company Name

Company Address

Company Number Phone

Company Logo

Choose File

No file chosen

Back

Submit

2. Master Pesawat

a. Main Grid

accelist⁺

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard











Company

Plane

Flight Schedule

Plane

Insert

Plane Name	Company Address	Plane Type	Plane Capacity	Plane Fuel Capacity	Action
Boeing 737	Boeing	300 passanger	Passanger	3000 litre	 
Boeing 737	Boeing	300 passanger	Passanger	3000 litre	 
Boeing 737	Boeing	300 passanger	Passanger	3000 litre	 
Boeing 737	Boeing	300 passanger	Passanger	3000 litre	 
Boeing 737	Boeing	300 passanger	Passanger	3000 litre	 

FirstLast

b. Form

accelist⁺

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard

Company

Plane

Flight Schedule

Form Data

Plane Name

Company Name

Plane Type

Plane Capacity

Plane Fuel Capacity

BackSubmit

3. Jadwal Penerbangan

a. Main Grid

accelist¹

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard











Company

Plane

Flight Schedule

Schedule

Insert

Plane Name	Company Name	Plane Capacity	Departure Date	Arrival Date	Gate Name	Action
Boeing 737	Boeing	300 passanger	30-March-2021 12:00	30-March-2021 13:00	Gate 1	 
Boeing 737	Boeing	300 passanger	30-March-2021 12:00	30-March-2021 13:00	Gate 1	 
Boeing 737	Boeing	300 passanger	30-March-2021 12:00	30-March-2021 13:00	Gate 1	 
Boeing 737	Boeing	300 passanger	30-March-2021 12:00	30-March-2021 13:00	Gate 1	 
Boeing 737	Boeing	300 passanger	30-March-2021 12:00	30-March-2021 13:00	Gate 1	 

PreviousNext

b. Form

accelist¹

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard

Company

Plane

Flight Schedule

Form Data

Company Name

Plane Name

Gate

Departure

Arrival

Departure Date

Arrival Date

Distance

Choose Plane

Choose Gate

dd/mm/yyyy

dd/mm/yyyy

Back

Submit

4. Dashboard

a. Main Grid

accelist⁺

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard

Company

Plane

Flight Schedule

Dashboard

Company Name	Plane Name	Total Flight	Action
Boeing 737	Boeing	321	<button>View Detail</button>
Boeing 737	Boeing	321	<button>View Detail</button>
Boeing 737	Boeing	321	<button>View Detail</button>
Boeing 737	Boeing	321	<button>View Detail</button>
Boeing 737	Boeing	321	<button>View Detail</button>

FirstLast

b. View Detail

accelist⁺

HomeFeaturesPricingFAQAbout

Signed in as Eagar

Dashboard

Company

Plane

Flight Schedule

Detail View

Plane Name	Plane Schedule	Total Flight
Boeing 737	30-March-2021	11
Boeing 737	30-March-2021	11
Boeing 737	30-March-2021	11
Boeing 737	30-March-2021	11
Boeing 737	30-March-2021	11

PrevNext

Back