

# Érettségi – GroupBy feladatok

---

## Alapfelállás

---

- Készíts WebAPI projektet: pl parancssorban `dotnet new webapi -n ErettsegiApi` vagy IDE segítségével
- Tesztadat (CSV) → SQLite: hozz létre egy `rettsegi.db` adatbázist, benne egy táblát `ExamResults` néven.
- Táblaséma javaslat:
  - Id (INTEGER, PK)
  - Name (TEXT)
  - Class (TEXT)
  - Subject (TEXT)
  - Grade (INTEGER)
- Scaffoldold a DbContextet és az entityt EF Core segítségével.
- Hozz létre controllert (pl. `ExamResultsController` ).
- A DbContextet példányosítsd a controllerben (new).

A tesztadat alapján 20 sor van, osztályok: 13.b, 13.c;

tantárgyak: Magyar, Matematika;

jegyek: 2–5.

---

## 1) Tantárgyankénti átlag

---

### Feladat

Készíts végpontot, ami tantárgyanként visszaadja az átlagjegyet.

### URL

GET /api/examresults/avg-by-subject

### SQL

```
SELECT
    Subject,
    ROUND(AVG(Grade), 2) AS AvgGrade
FROM ExamResults
GROUP BY Subject
ORDER BY Subject;
```

## Sikeres válasz (minta)

```
{
  "data": [
    { "subject": "Magyar", "avgGrade": 3.4 },
    { "subject": "Matematika", "avgGrade": 3.7 }
  ]
}
```

## 2) Osztályonként hány különböző diák van

### Feladat

Add vissza osztályonként a különböző tanulók számát.

### URL

GET /api/examresults/student-count-by-class

### SQL

```
SELECT
    Class,
    COUNT(DISTINCT Name) AS StudentCount
FROM ExamResults
GROUP BY Class
ORDER BY Class;
```

## Sikeres válasz (minta)

```
{  
  "data": [  
    { "class": "13.b", "studentCount": 5 },  
    { "class": "13.c", "studentCount": 5 }  
  ]  
}
```

### 3) Osztályátlag

#### Feladat

Add vissza osztályonként az átlagjegyet.

#### URL

GET /api/examresults/avg-by-class

#### SQL

```
SELECT  
  Class,  
  ROUND(AVG(Grade), 2) AS AvgGrade  
FROM ExamResults  
GROUP BY Class  
ORDER BY Class;
```

### Sikeres válasz (minta)

```
{  
  "data": [  
    { "class": "13.b", "avgGrade": 4.0 },  
    { "class": "13.c", "avgGrade": 3.1 }  
  ]  
}
```

### 4) Osztály + tantárgy bontású átlag

## Feladat

Add vissza osztályonként és tantárgyanként az átlagjegyet.

## URL

GET /api/examresults/avg-by-class-and-subject

## SQL

```
SELECT
    Class,
    Subject,
    ROUND(AVG(Grade), 2) AS AvgGrade
FROM ExamResults
GROUP BY Class, Subject
ORDER BY Class, Subject;
```

## Sikeres válasz (minta)

```
{
  "data": [
    { "class": "13.b", "subject": "Magyar", "avgGrade": 3.8 },
    { "class": "13.b", "subject": "Matematika", "avgGrade": 4.2 },
    { "class": "13.c", "subject": "Magyar", "avgGrade": 3.0 },
    { "class": "13.c", "subject": "Matematika", "avgGrade": 3.2 }
  ]
}
```

## 5) Jegyelőfordulás tantárgyanként

## Feladat

Add vissza tantárgyanként, hogy hány darab 2/3/4/5-ös jegy van.

## URL

GET /api/examresults/grade-distribution-by-subject

## SQL

```
SELECT
    Subject,
    Grade,
    COUNT(*) AS Count
FROM ExamResults
GROUP BY Subject, Grade
ORDER BY Subject, Grade;
```

## Sikeres válasz (minta)

```
{
  "data": [
    { "subject": "Magyar", "grade": 2, "count": 3 },
    { "subject": "Magyar", "grade": 3, "count": 2 },
    { "subject": "Magyar", "grade": 4, "count": 3 },
    { "subject": "Magyar", "grade": 5, "count": 2 },
    { "subject": "Matematika", "grade": 2, "count": 4 },
    { "subject": "Matematika", "grade": 4, "count": 1 },
    { "subject": "Matematika", "grade": 5, "count": 5 }
  ]
}
```