

## SAP ABAP Programozás Alapjai – Nappali

### Második gyakorlat

A gyakorlaton elkészített munkáját **jegyzőkönyvezzé**, amely kiterjed a meghozott fejlesztői döntések, az elkészített forráskód, a futtatás eredményeként kapott konzol kimenetek dokumentálására!

#### 1. feladat – ABAP Dictionary modellezés hallgatói szemeszter adatokhoz

A feladat célja, hogy a hallgató megismerje, hogyan kell adatmodelleket létrehozni az ABAP Dictionary-ben az ABAP on Cloud környezetben.

Ebben a feladatban:

- domain-eket
- data elementeket
- struktúrákat
- adatbázis táblákat

kell létrehozni.

##### 1.1. Domain-ek létrehozása

**Hozza létre a következő domain-eket:**

Domain	Jelentés	Példa típus
Student Neptun Code	hallgató azonosító	CHAR(6)
Enrollment Date	Képzés kezdete	DATN vagy DATS
Grade	jegy	NUMC(1) vagy INT

A grade domain esetén adjon meg megengedett értékészletet is pl.: **0 – 5**

##### 1.2. Data Element-ek létrehozása

A domain-ekhez hozzon létre data elementeket. Minden data elementnél töltsse ki:

- Short text
- Medium text
- Long text

Példa

Data Element	Domain
ZSTUDENT_NEPTUN	Student Neptun domain
ZCOURSE_CODE	Course Code domain
ZSEMESTER_ID	Semester ID domain

ZGRADE	Grade domain
--------	--------------

Hozzon létre data elementeket a következő mezőkhöz is:

- student\_name
- enrolment\_date
- semester\_name

Használjon beépített adattípusokat.

### Opcionális feladat - Dokumentáció

Legalább egy data elementhez írjon dokumentációt. Példa:

Neptun code uniquely identifies a student in the university system.

### 1.3. Struktúrák létrehozása

Hozd létre a következő struktúrákat:

**Course record**, az alábbi példa mezőkkel

- course\_code
- course\_name
- grade

**Semester record**, az alábbi példa mezőkkel

- semester\_id
- semester\_name

**Student record**, az alábbi példa mezőkkel

- student\_neptun\_code
- student\_name
- enrolment\_date

### 1.4. Adatbázis táblák létrehozása

Hozzon létre adatbázis táblákat.

**Student table, Táblanév példa: Z<NEPTUN>\_STUD**

Mezők:

Field	Key
client	X
student_neptun_code	X
student_name	
enrolment_date	

---

**Student semester table, Táblanév Z<NEPTUN>\_SMS**

Mezők:

Field	Key
client	X
student_neptun_code	X
semester_id	X
semester_name	

**Course grade table, Táblanév Z<NEPTUN>\_CRS**

Mezők:

Field	Key
client	X
student_neptun_code	X
semester_id	X
course_code	X
grade	

**Exam table, Táblanév Z<NEPTUN>\_EXAM**

Mezők:

Field	Key
client	X
exam_id	X
student_neptun	
course_code	
semester_code	
grade	

## 2. feladat – Adatbázis feltöltése ABAP alkalmazással

A feladat célja, hogy egy olyan ABAP konzolban futtatható alkalmazást készítsen, amely tesztadatokat tölt az adatbázisba.

A program elvárt működése:

A program minden futáskor:

1. törölje a korábbi tesztadatokat
2. tölts fel új adatokkal a táblákat

Az alkalmazás hozzon létre:

- Legalább 3 hallgatót, például:
  - ABC123
  - DEF456
  - GHI789
- Minden hallgatóhoz legalább 2 szemesztert, például
  - 2023\_1
  - 2023\_2
- Minden szemeszterhez több kurzust, például
  - ABAP
  - DATABASE
  - ALGORITHMS

Fontos adatbázis műveletek: INSERT, vagy MODIFY

Hibakezelés

Minden adatbázis művelet után ellenőrizze a sy-subrc értékét és hiba esetén írjon “beégetett” üzenetet a konzolra, például:

IF sy-subrc <> 0.

WRITE: / 'Insert failed'.

ENDIF.

### 3. feladat: Modularizáció Function Group és Function Module segítségével

A feladat célja gyakorolni, hogyan lehet az újrafelhasználható üzleti logikát készíteni Function Groupok és function module-ok segítségével. A feladat a létrehozott adatstruktúrákra és adatbázis táblákra épül.

#### 4.1. Function Group létrehozása vizsgák kezeléshez

##### Cél

Ez a Function Group tartalmazza a hallgatók kezelésével kapcsolatos funkciókat.

Hozzon létre egy új Function Groupot:

- Név: Z<neptun>\_EXAM\_MGMNT
- Leírás: Function Group for Student Management

Aktiválás után ellenőrizze a generált objektumokat:

- fő program
- TOP include
- egyéb include-ok

#### 4.2. Function Module-ok létrehozása

A Function Groupon belül hozzon létre három Function Module-t.

##### Function Module 1: Z<neptun>\_GET\_SEMESTER

A function module célja meglévő szemeszter lekérdezése az adatbázisból

- **Import paraméterek**
  - IV\_SEMESTER\_CODE (kötelező)
- **Export paraméterek**
  - ES\_SEMESTER\_DATA (struktúra)
- **Exceptions**
  - SEMESTER\_DOES\_NOT\_EXIST

##### Function Module 2: Z<neptun>\_GET\_STUDENT

A function module célja létező hallgató lekérdezése az adatbázisból.

- **Import paraméter**
  - IV\_NEPTUN\_CODE
- **Export paraméter**
  - ES\_STUDENT\_DATA
- **Exceptions**
  - STUDENT\_DOES\_NOT\_EXIST

#### **4. feladat: Funkciós modulok használata ABAP console alkalmazásban**

A meglévő minta adatbázist létrehozó console alkalmazás funkcióját valósítsa meg egy új alkalmazásban, amely a létrehozott funkciós modulokat használja.