

○ 02.25 수 ○

RAP

멘토링 3회차

ABAP 데이터 구조 설계와 CDS 개발

멘토 김예원

• INDEX

목차

01

CDS View 생성 방법 비교 (View vs View Entity)

02

Annotation Propagation: 상속되는 설정값

03

ABAP Dictionary 외래키(Foreign Key) 설정

04

CDS Association

05

CDS WHERE·필터·집계 함수

01

SAP GUI가 있는데 왜 Eclipse를 배우는가?

- 최신 기술은 Eclipse에서만 가능. SAP GUI에서는 개발 불가 또는 제한적.
- CDS View, RAP, ABAP Cloud 등등 ..
- ADT = ABAP Development Tools : Eclipse에서 ABAP 개발하게 해주는 SAP 공식 개발 도구

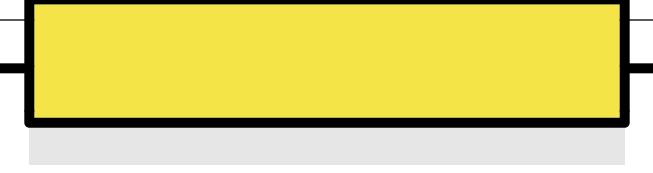
CDS View란?

- 데이터를 조회하기 위해 미리 설계하는 가상 데이터 구조
- DB 테이블을 기반으로 조회 구조를 정의해두는 방식

```
SELECT * FROM sflight INTO TABLE lt_data.
```

```
@AbapCatalog.sqlViewName: 'ZCDSSCARR_B03_S'  
@AbapCatalog.compiler.compareFilter: true  
@AbapCatalog.preserveKey: true  
@AccessControl.authorizationCheck: #NOT_REQUIRED  
@EndUserText.label: 'Scarr SCDS View'  
@Metadata.ignorePropagatedAnnotations: true  
define view ZCDSSCARR_B03 as select from scarr  
{  
    key carrid as Carr_Id,  
    carrname as Carr_Name,  
    currcode,  
    url  
}
```

→ CDS View는 가상 데이터 구조를 정의하면서 동시에 SELECT가 가능



01

CDS 금액·수량 Semantics 설정

→ 이 숫자가 그냥 숫자인지, 돈인지, 수량인지를 알려주는 표시

구문법 (*define view*)

→ 데이터 타입(DDIC) 정보가 자동으로 넘어오지 않아 `@Semantics.currencyCode: true` 등을
직접 명시해야 함

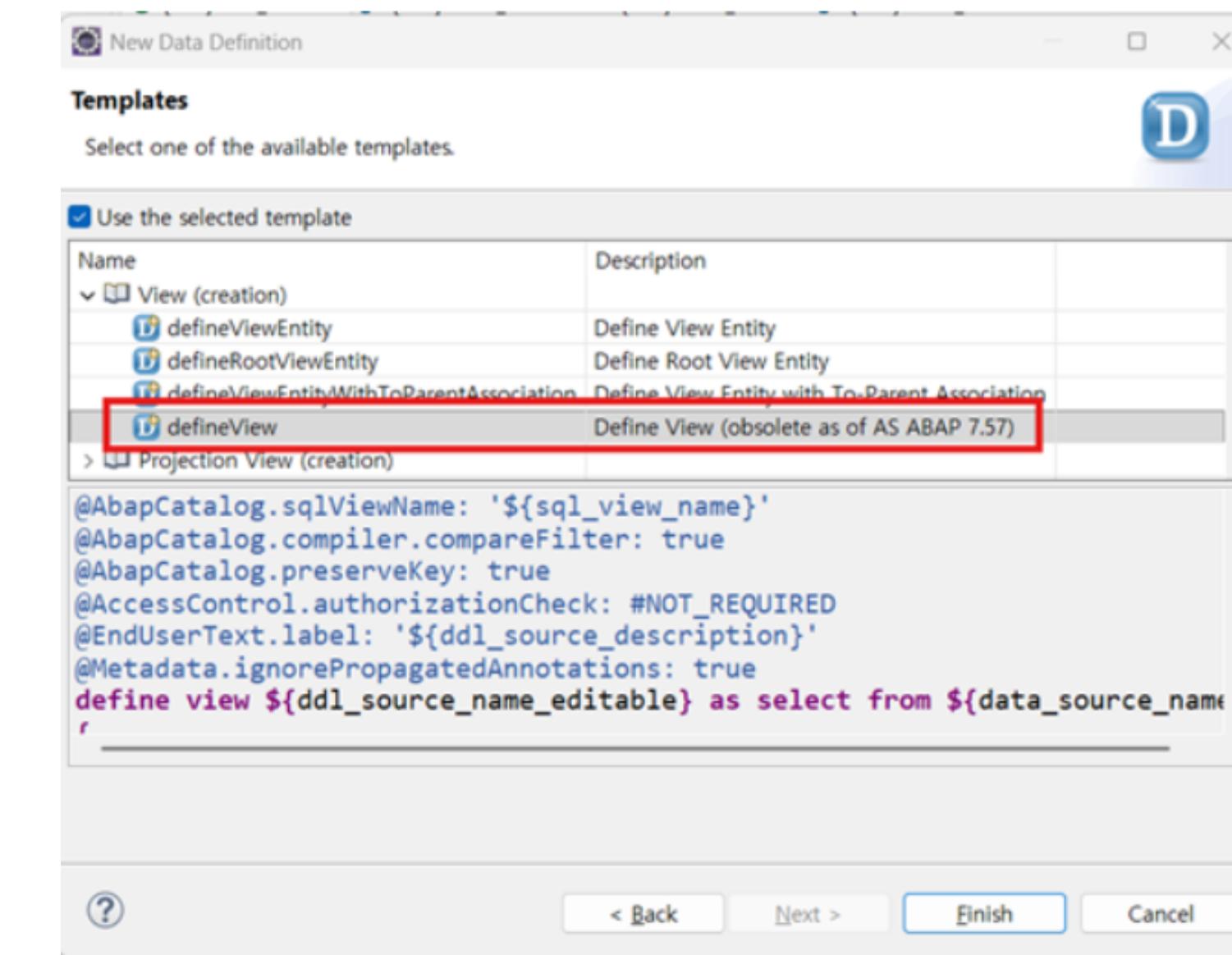
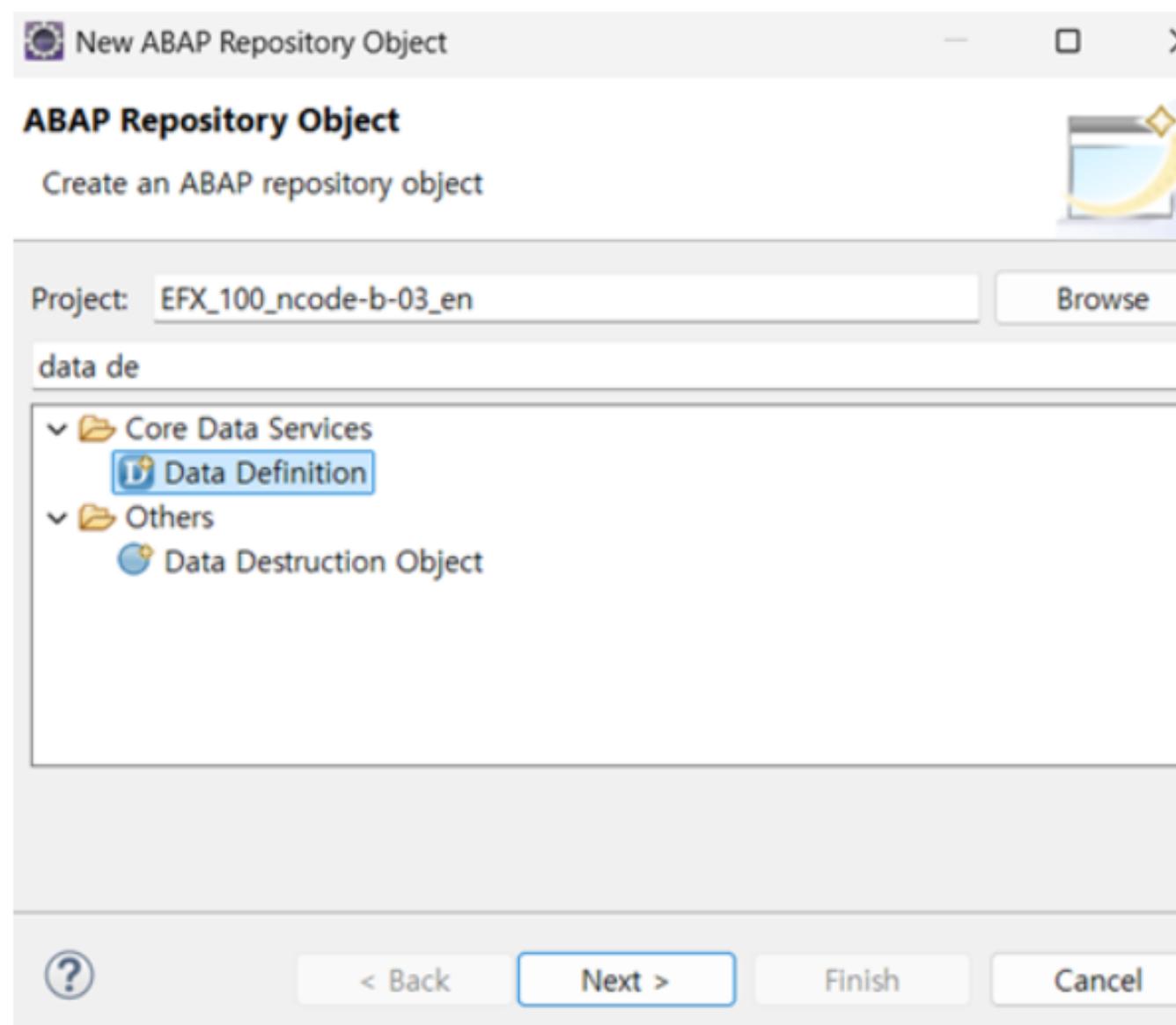
신문법 (*define view entity*)

→ 원본 테이블의 참조 정보를 자동으로 상속받으므로 보통 생략 가능

01

CDS View 생성 방법 비교 (View vs View Entity) → 구문법

→ 테이블에서 필요한 데이터만 골라서, 화면이나 프로그램에서 바로 사용할 수 있게 정리해놓은 것



01

CDS View 생성 방법 비교 (View vs View Entity)

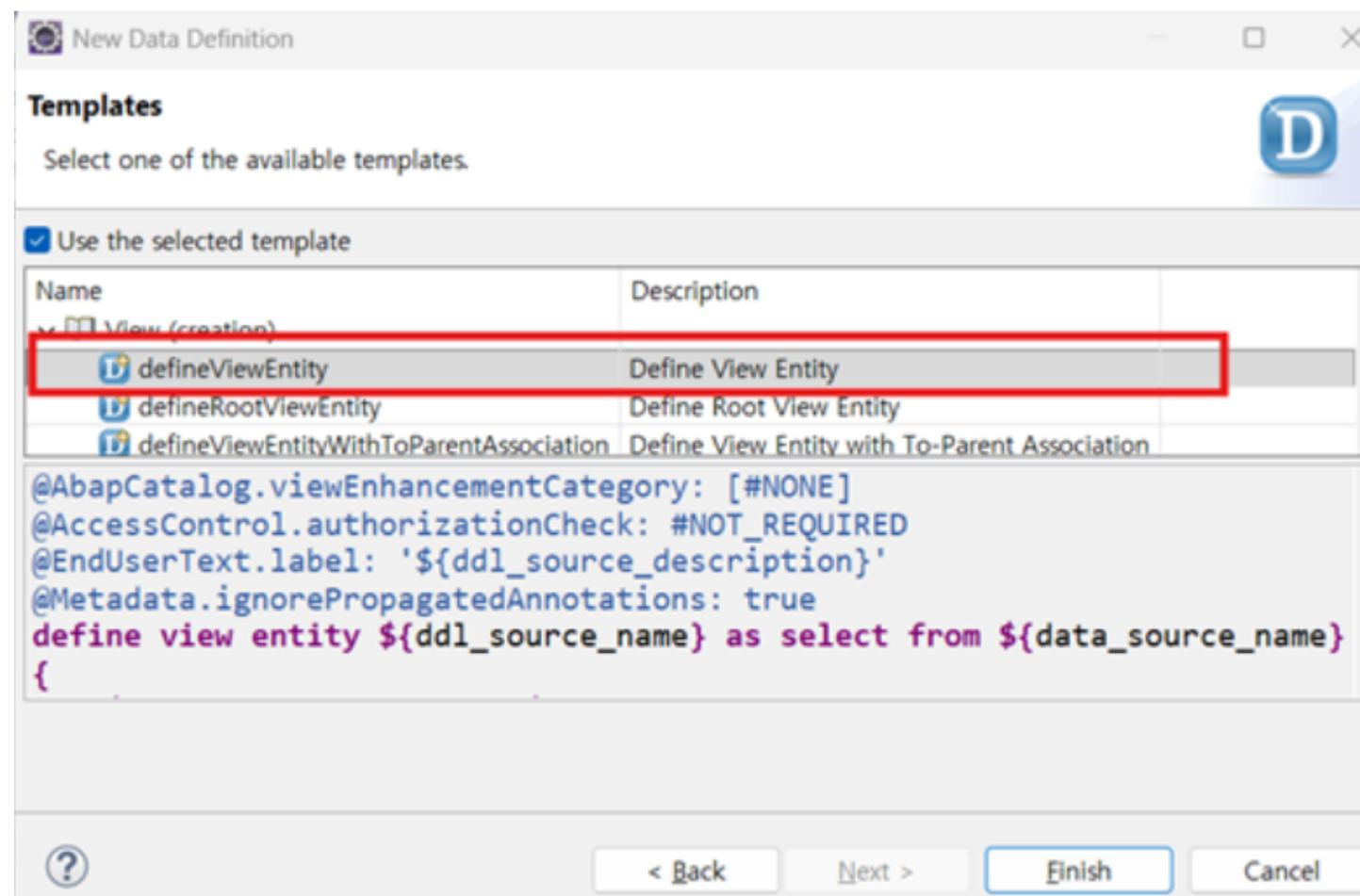
```
@AbapCatalog.sqlViewName: 'ZCDSFLT2_B03'  
@AbapCatalog.compiler.compareFilter: true  
@AbapCatalog.preserveKey: true  
@AccessControl.authorizationCheck: #NOT_REQUIRED  
@EndUserText.label: 'Flight CDS View 2'  
@Metadata.ignorePropagatedAnnotations: true  
define view ZCDSFLIGHT2_B03 as select from sflight  
{  
    key carrid as Carrid,  
    key connid as Connid,  
    key fldate as Fldate,  
  
    @Semantics.amount.currencyCode: 'Currency'  
    price as Price,  
    @Semantics.currencyCode: true  
    currency as Currency,  
    planetype as Planetype,  
    seatsmax as Seatsmax,  
    seatsocc as Seatsocc,  
    @Semantics.amount.currencyCode: 'Currency'  
    paymentsum as Paymentsum  
}
```

→ DB에 생성될 SQL View 이름 지정 (구문법에서 필수)
→ 조회할 때 조건(WHERE절)을 더 효율적으로 비교하도록 하는 설정
→ 원본 테이블의 기본키를 그대로 유지
→ 이 View는 권한 체크 없이 누구나 조회 가능하게 설정
→ 화면에서 보여질 설명 이름 (라벨)
→ 하위 테이블이나 View에서 넘어오는 Annotation 설정을 무시

→ 숫자는 돈(금액)이고 단위는 Currency 필드에 있음
→ 이 필드는 통화코드(USD, KRW 등) 임을 표시

01

CDS View 생성 방법 비교 (View vs View Entity) → 신문법



```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Flight CDS View'
@Metadata.ignorePropagatedAnnotations: true

define view entity ZCDSFLIGHT_B03 as select from sflight
{
    key carrid as Carrid,
    key connid as Connid,
    key fldate as Fldate,
    @Semantics.amount.currencyCode: 'Currency'
    price as Price,
    currency as Currency,
    planetype as Planetype,
    seatsmax as Seatsmax,
    seatsocc as Seatsocc,
    @Semantics.amount.currencyCode: 'Currency'
    paymentsum as Paymentsum
}
```

02

Annotation Propagation: 상속되는 설정값

- 하위 뷰(부모)에서 설정한 필드 속성(@)은 상위 뷰(자식)가 그대로 물려받아 사용 가능
- 전파 O(필드 애노테이션): 금액 설정이나 라벨처럼 특정 필드에 붙인 설정
전파 X(뷰 애노테이션): 권한 체크나 뷰 전체 설명 같이 View 맨 윗부분에 적힌 설정

02

Annotation Propagation: 상속되는 설정값

```
1 @AbapCatalog.viewEnhancementCategory: [#NONE]
2 @AccessControl.authorizationCheck: #NOT_REQUIRED
3 @EndUserText.label: 'Flight CDS View'
4 @Metadata.ignorePropagatedAnnotations: true
5 define view entity ZCDSFLIGHT_B03 as select from sflight
6 {
7   key carrid as Carrid,
8   key connid as Connid,
9   key fldate as Fldate,
10
11  @Semantics.amount.currencyCode: 'Currency'
12  price as Price,
13  currency as Currency,
14  planetype as Planetype,
15  seatsmax as Seatsmax,
16  seatsocc as Seatsocc,
17  @Semantics.amount.currencyCode: 'Currency'
18  paymentsum as Paymentsum
19 }
20
```

< 하위 View (부모) >

```
1 @AbapCatalog.viewEnhancementCategory: [#NONE]
2 @AccessControl.authorizationCheck: #NOT_REQUIRED
3 @EndUserText.label: 'Flight CDS Propagated'
4 @Metadata.ignorePropagatedAnnotations: false
5 define view entity ZCDSFLIGHT_PROPA_B03 as select from ZCDSFLIGHT_B03
6 {
7   key Carrid,
8   key Connid,
9   key Fldate,
10  Price,
11  Currency,
12  Planetype,
13  Seatsmax,
14  Seatsocc,
15  Paymentsum
16 }
17
```

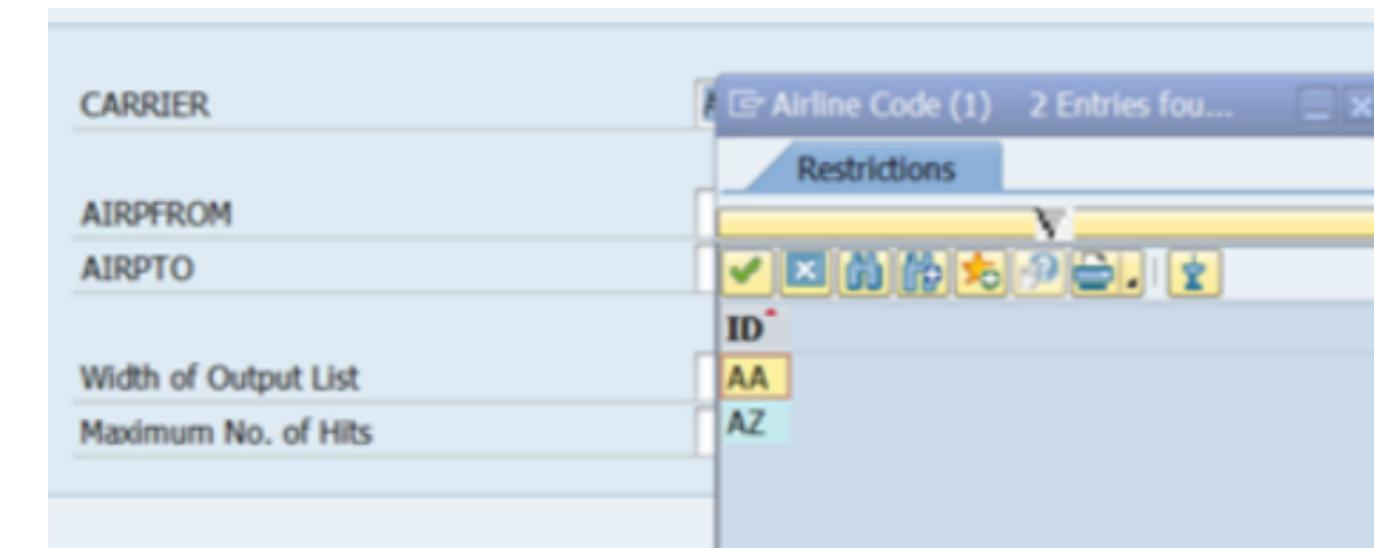
< 상위 View (자식) >

03

ABAP Dictionary 외래키(Foreign Key) 설정

- 체크 테이블은 이 학교의 전체 학생 명단
- 외래키는 동아리 가입 신청서를 받을 때 이 학교의 명단에 있는 학생인지 대조해보는 검사 규칙

```
[ForeignKey Table(Source), Check Table(Target)]  
with foreign key [0..*, 1]
```



- 왼쪽 (Source/내 테이블) : 최소 0건에서 최대 여러 건(*)의 데이터가 존재할 수 있음
오른쪽 (Target/체크 테이블) : 최대 1건의 기준 데이터가 존재 (0..1)

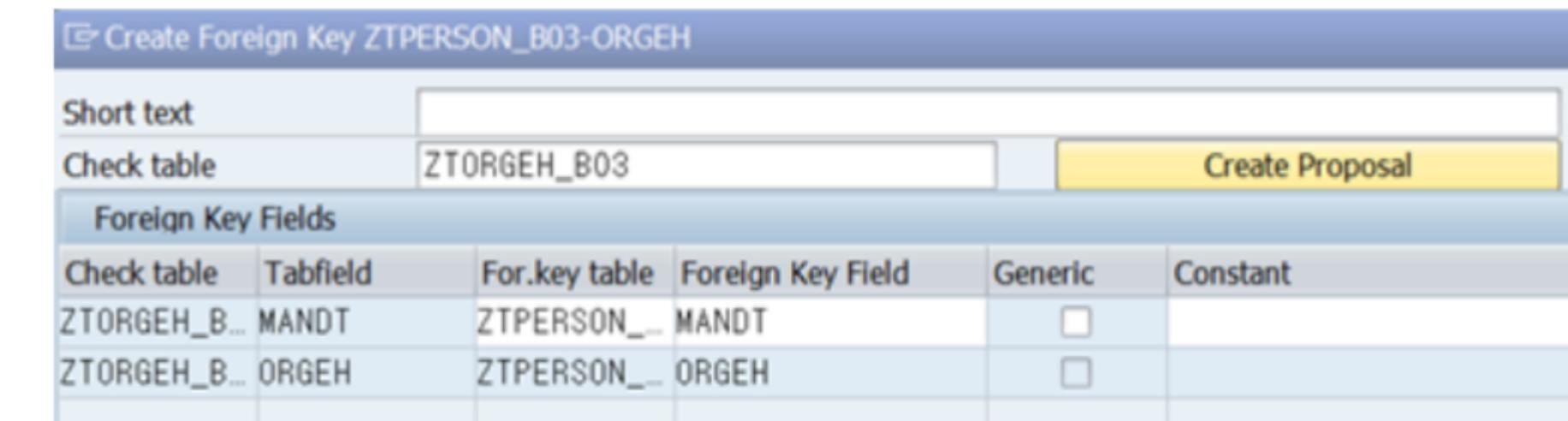
여러 명의 직원(Source)이 하나의 부서(Target)에 소속된다

03

ABAP Dictionary 외래키(Foreign Key) 설정

```
1 @EndUserText.label : 'Department (with Foreign Key Relations)'
2 @AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #RESTRICTED
6 define table z03depment {
7
8   key client    : abap.clnt not null;
9   key id        : /lrn/department_id not null;
10  description   : /lrn/depment_description;
11  @AbapCatalog.foreignKey.screenCheck : false
12  head_id      : /lrn/depment_head
13  with foreign key [0..1,0..1] /lrn/employ_rel
14    where client = z03depment.client
15    and employee_id = z03depment.head_id;
16  @AbapCatalog.foreignKey.screenCheck : false
17  assistant_id : /lrn/depment_assistant
18  with foreign key [0..1,1] /lrn/employ_rel
19    where client = z03depment.client
20    and employee_id = z03depment.assistant_id;
21  include /lrn/s_admin;
22
23 }
```

```
define table /lrn/employ_rel {
  key client    : abap.clnt not null;
  key employee_id : /lrn/employee_id not null;
```



04

CDS Association

- 테이블끼리 미리 연결 통로만 만들어두고, 실제로 데이터가 필요할 때만 알아서 Join
- Join은 즉시 결합하고, Association은 관계만 정의 후 필요 시 조회함

\$projection → 현재 뷰의 결과물 필드를 가리킴 (별칭)

[min..max] (Cardinality)

→ Target 테이블에서 ON 조건에 따라 매칭했을 때, 반환될 수 있는 결과 행의 최소~최대 개수를 의미

```
association [min..max] to TargetView as _Assoc
```

04

CDS Association

1. 타겟의 PK가 1개 일 때

- [0..1] 또는 [1..1] (최대 1건): 타겟의 PK를 조건에 걸었을 때
- [0..*] 또는 [1..*] (여러 건): 타겟의 PK가 아닌 일반 필드를 조건에 걸었을 때

2. 타겟의 PK가 2개 이상 일 때

- [0..1] 또는 [1..1] (최대 1건): 타겟의 PK 전체를 다 조건에 걸었을 때
- [0..*] 또는 [1..*] (여러 건): 타겟의 상대방의 PK 중 일부 또는 일반필드를 조건에 걸었을 때

04

CDS Association

```
define view entity R_Employee
  as select from employee

  association [0..*] to R_Department as _Department
    on $projection.DepartmentId = _Department.Id
{
  key employee_id as EmployeeId,
    department_id as DepartmentId
}
```

```
define view entity R_Department
  as select from department
{
  key id as Id,
    department_designation as DepartmentDesignation
}
```

왜 *Syntax Warning!* 뜰까?

04

CDS Association

1) Target Entity(zdept)의 PK가 'dept_id' 하나일 때 카드널리티는?

```
define view entity ZI_Employee as select from zemp
  association [ ? ] to zdept as _Department
    on $projection.DeptCode = _Department.dept_id
  {
    key emp_no as EmployeeNumber,
    dept_code as DeptCode
  }
```

- 1) [0..1]
- 2) [1..1]
- 3) [0..*]
- 4) [1..*]

04

CDS Association

2) scarr의 PK는 'carrid'이다. 시스템 경고(Warning)가 발생하는 이유는 ?

```
association [0..*] to scarr as _Carrier  
on $projection.CarrierId = _Carrier.carrid
```

04

CDS Association

3) Target Entity(*sap_planes*)의 PK가 *CARRID*, *PLANE_ID* 일 때 카디널리티는?

```
define view entity ZI_FlightInfo
    as select from spfli
        association [ ? ] to sap_planes as _Plane
            on $projection.CarrierId = _Plane.CARRID
{
    key carrid as CarrierId,
    key connid as ConnectionId,
    _Plane
}
```

- 1) [0..1]
- 2) [1..1]
- 3) [0..*]
- 4) [1..*]

04

CDS Association - 실제 JOIN 발생 조건

1. Join이 발생하지 않는 경우

```
define view entity C_Employee
  as select from R_Employee
    association [0..1] to R_Department as _Department
      on $projection.DepartmentId = _Department.Id
{
  key EmployeeId,           // 직원 ID (PK)
        FirstName,          // 이름
        LastName,            // 성
        DepartmentId,        // 부서 ID

  _Department             // 조인 발생 X
}
```

2. Join이 발생하는 경우

```
define view entity C_Employee
  as select from R_Employee
    association [0..1] to R_Department as _Department
      on $projection.DepartmentId = _Department.Id
{
  key EmployeeId,           // 직원 ID (PK)
        FirstName,          // 이름
        LastName,            // 성
        DepartmentId,        // 부서 ID

  _Department.DependentDesignation // 부서 테이블
}
```

05

CDS WHERE·필터·집계 함수

1) 기준 테이블 필드

→ 재직 중인 사원들만 보기

2) 1:1 Association 필드

→ 부서 a 소속인 사원들만 보기

3) 1:N Association 필드

→ 사원 한 명은 여러 개의 부서

→ 특정 사원만 줄이 늘어남 !!

```
5 define view entity ZCDSSPFLI_B03
6   as select from spfli as sf
7   // association 생략 시 기본값은 [0..1]
8   association [1..1] to ZCDSSCARR2_B03 as _Carrier
9   // on $projection.Carr_id = _Carrier.Carrid
10  on sf.carrid = _Carrier.Carrid
11  association [*] to ZCDSTFLIGHT2_B03 as _flight // [0..*]
12  on sf.carrid = _flight.Carrid
13  and sf.connid = _flight.Connid
14 {
15  key carrid
16  key connid
17  countryfr
18  countryto
```

```
           as Carr_id,
           as Connid,
           as Countryfr,
           as Countryto,
```

```
31
32 }
33 // where carrid = 'AA'      // where절 옆 필드는 db 필드 기준
34 // where _flight.Seatsocc > 100 // 카디널리티가 *로 설정되어 있으므로 사용 불가
35 where
36   _Carrier.Carrid = 'AA'
```

05 CDS WHERE·필터·집계 함수

- 집계 함수(min, max, sum, count 등)는 여러 줄(*)의 데이터를 딱 한 줄로 압축하기 위해 사용
- 집계 대상을 가져오는 Association은 반드시 [*] (1:N) 관계여야 의미있음
- 집계 함수를 적용하지 않은 모든 일반 선택 필드는 **group by** 절에 적어줘야 함
- count(**distinct** ...) 처럼 사용하며 중복된 값은 1개로 처리

```
11  association [*]      to ZCDSFLIGHT2_B03 as _flight // [0..*]
12  on   sf.carrid = _flight.Carrid
13  and sf.connid = _flight.Connid

22  min( _flight[ Seatsocc = Seatsmax ].Fldate ) as minFlight,
23  max( _flight.Fldate ) as maxFlight,
24  count( distinct _flight.Fldate ) as cntFlight,
25
```

```
32  }
33  // where carrid = 'AA'
34  // where _flight.Seatsocc
35  where
36  Carrier.Carrid = 'AA'
37  group by
38  carrid,
39  connid,
40  countryfr,
41  countryto
42
```

05

CDS WHERE·필터·집계 함수

```
1@AbapCatalog.viewEnhancementCategory: [#NONE]
2@AccessControl.authorizationCheck: #NOT_REQUIRED
3@EndUserText.label: 'Scarr CDS View 2'
4@Metadata.ignorePropagatedAnnotations: true
5define view entity ZCDSSCARR2_B03 as select from scarr
6    association[0..*] to I_CurrencyText as _Text
7        on $projection.CurrCode = _Text.Currency
8    association[*] to ZCDSSPFLI_B03 as _spfli
9        on scarr.carrid = _spfli.Carr_id
10{
11    key carrid as Carrid, // 주석은 이렇게
12        Carrname as Carrname, // 주석 !!
13        currCode as Currcode,
14
15//      _spfli,
16//      _Text
17        _Text[ 1: Language = $session.system_language ].CurrencyName as CurrName // EN
18/*
19    다중 주석 처리
20    Ctrl + Shift + < : 주석 적용
21    Ctrl + Shift + > : 주석 해제
22*/
23}
24
```