

CDS VIEW 를 이용한 odata service 만들기

odata servie 를 기존의 ddic 를 사용해서만들어도 된다.

하지만 이러면 CRUD 전체를 다 구현해야 한다.

그리고 association 을 설정하는 것이 복잡하다.

그리고 추가로 query option 들을 다 구현해야한다(top,skip,count 는 구현되어 있음).

그래서 쉬운 방법으로 CDS View 를 이용하여 odata service를 만드는 것을 추천한다.

CDS View 를 사용할경우 read 와 query는 구현이 되어 있다.

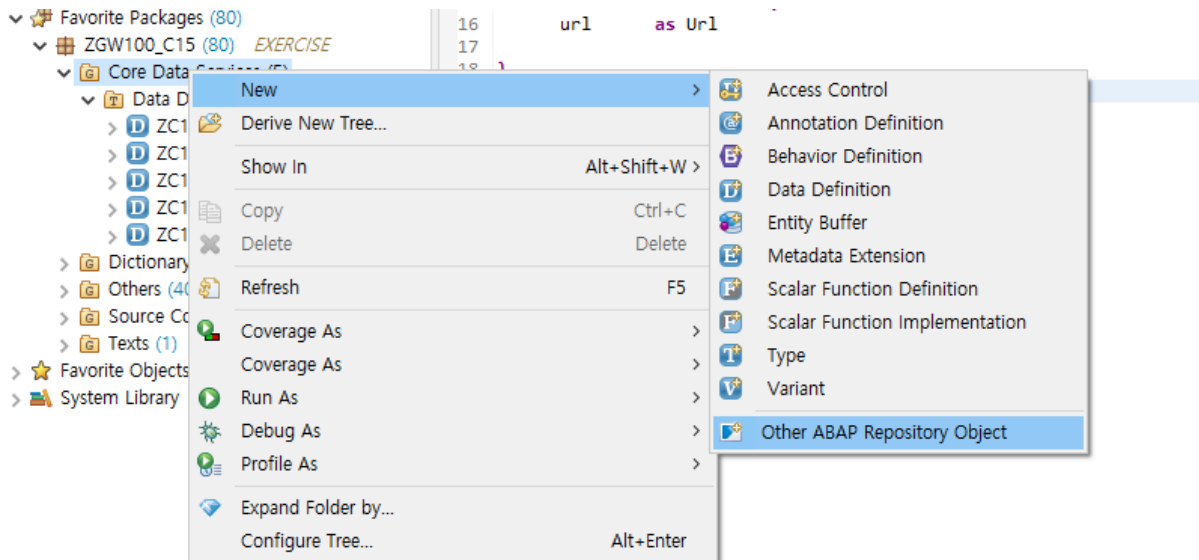
그리고 추가로 query option 들이 다 구현되어 있다.

왜??? 우리가 cdw view 를 만들때 select 문을 적으니까 따로 implementation 에거 구현을 할 필요가 없다.

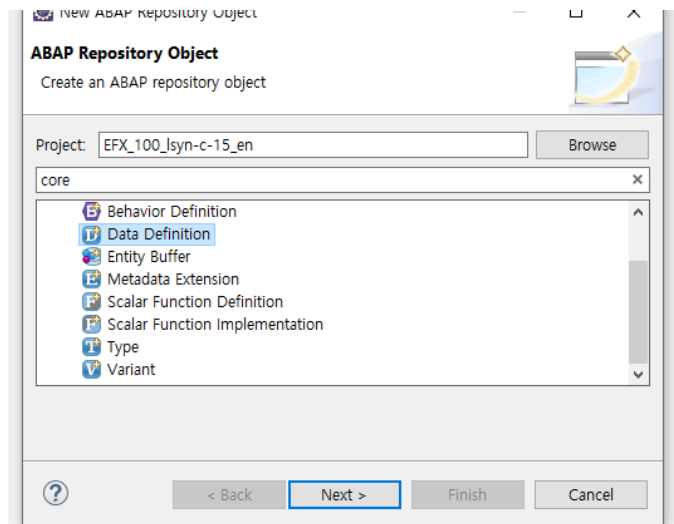
Step.1

이클립스 상에서 CDS View 를 만들어라..

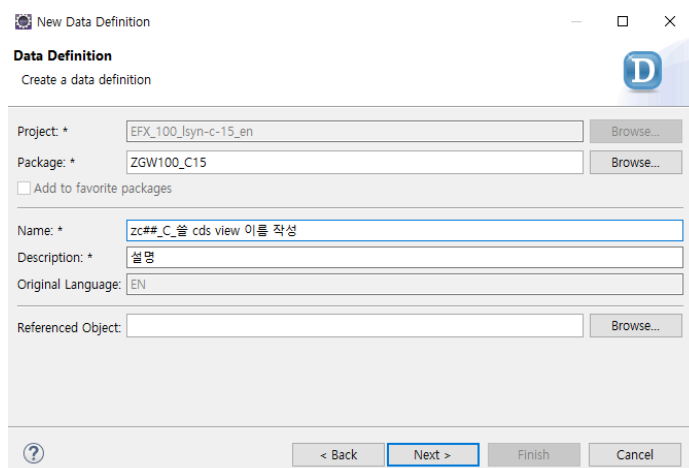
기본 양식은 다음과 같다.



패키지 우클릭 → new → other abap repository object



core 를 입력하고 → data definition 을 선택후 next



사용할 cds view 의 이름과 설명을 적고 next

1. Association을 주지 않고 View 한개만 사용할 경우 (navigation proeprty 를 사용 안할때, table 한개만 사용하려고 할때)

```
@AbapCatalog.sqlViewName: 'ZC15CSARR2' // GW 에서 DDIC 에서 조회할 V
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'SCARR'
@Metadata.ignorePropagatedAnnotations: true

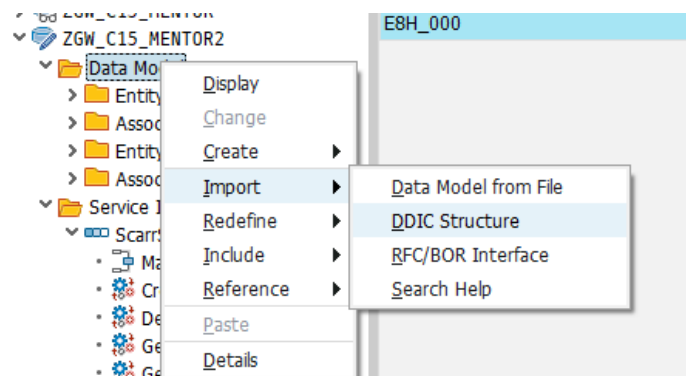
@OData.publish: true // oData service 를 자동으로 등록한다

define view ZC15_C_SARR_2 as select from scarr

{
  key carrid as Carrid,
  carrname as Carrname,
  currcode as Currcode,
  url as Url

}
```

사용할 Table을 사용하여 CDS View 를 만들어라.



위의 주석을 보라

@AbapCatalog.sqlViewName: '~~ ' 이 이름으로 CDS View 를 GW 에서 import 한다.

이름은 define view 의 이름을 붙여서 작성한다. ('_' 언더바를 제거하고)

2. Association을 줄 경우 (navigation property 를 사용 table 여러개를 연결하고 싶을 때)

```
@AbapCatalog.sqlViewName: 'ZC15CSARR' // GW 에서 DDIC 에서 조회할 VIEW
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'SCARR'
@Metadata.ignorePropagatedAnnotations: true

@OData.publish: true // oData service 를 자동으로 등록한다

define view ZC15_C_SARR
as select from scarr
association [*] to ZC15_C_SFLIGHT as _ZSFLIGHT on scarr.carrid = _ZSFLIGHT

{
  key carrid as Carrid,
    carrname as Carrname,
    currcode as Currcode,
    url as Url,

    /* exposed association */
    _ZSFLIGHT // _ZSFLIGHT 를 OData에서 네비게이션 프로퍼티로 노출
}
```

```
@AbapCatalog.sqlViewName: 'ZC15CSFLIGHT' // GW 에서 DDIC 에서 조회할 VIEW
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'sflight'
```

```
@Metadata.ignorePropagatedAnnotations: true
```

```
@OData.publish: true // oData service 를 자동으로 등록한다
```

```
define view ZC15_C_SFLIGHT
```

```
as select from sflight
```

```
association [1] to ZC15_C_SARR as _ZSCARR on $projection.Carrid = _ZSCA
```

```
{
```

```
key carrid as Carrid,
```

```
key connid as Connid,
```

```
key fldate as Fldate,
```

```
price as Price,
```

```
currency as Currency,
```

```
planetype as Planetype,
```

```
seatsmax as Seatsmax,
```

```
seatsocc as Seatsocc,
```

```
paymentsum as Paymentsum,
```

```
seatsmax_b as SeatsmaxB,
```

```
seatsocc_b as SeatsoccB,
```

```
seatsmax_f as SeatsmaxF,
```

```
seatsocc_f as SeatsoccF,
```

```
/* exposed association */
```

```
_ZSCARR // _ZSCARR 를 OData에서 네비게이션 프로퍼티로 노출
```

```
}
```

association 은 join 처럼 연결된 테이블에 내가 선택한 키값을 기준으로 데이터를 가져 오는 것이다.

그럼으로 association은 각 테이블의 key field를 기준으로 join 을 하듯 작성해야한다,

Association을 작성하는 방법

```
define view ZC15_C_SARR
```

```
as select from scarr
```

```
association [*] to ZC15_C_SFLIGHT as _ZSFLIGHT on scarr.carrid = _ZSFLIGHT
```

association[연결 관계] 기본적으로 0..n 의 양식이다 내 테이블이 0~1개 가 연결이 가능하다 그래서 0 은 기본값으로 생략을 하고 작성할 수 있다.

{[0..1] 이거랑 [1] 이랑 같다} - 0 대 1 관계,

[1..2] - 1대 2 관계

[0..*] - 0대 n 의 관계

내 view 기준으로 작성한다. 위의 내용을 풀이하면

scarr table 로 만든 ZC15_C_SARR view 는 ZC15_C_SFLIGHT view 와 carrid 를 key field 로 사용하여 carrid 가 같을 경우 [0 대 n] 의 관계를 가진다.

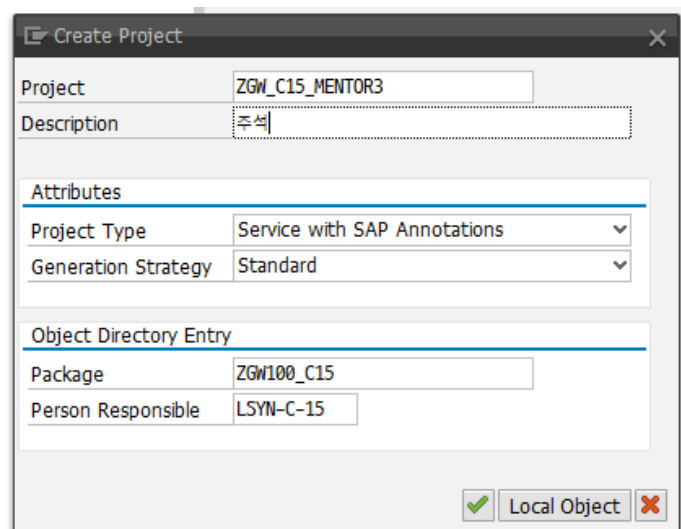
반대 쪽 view 도 똑같이 association 을 주면 된다.

그리고 항상 ACTIVE 를 해줘라!!!! 안해주면 SEGW 에서 조회가 안됨

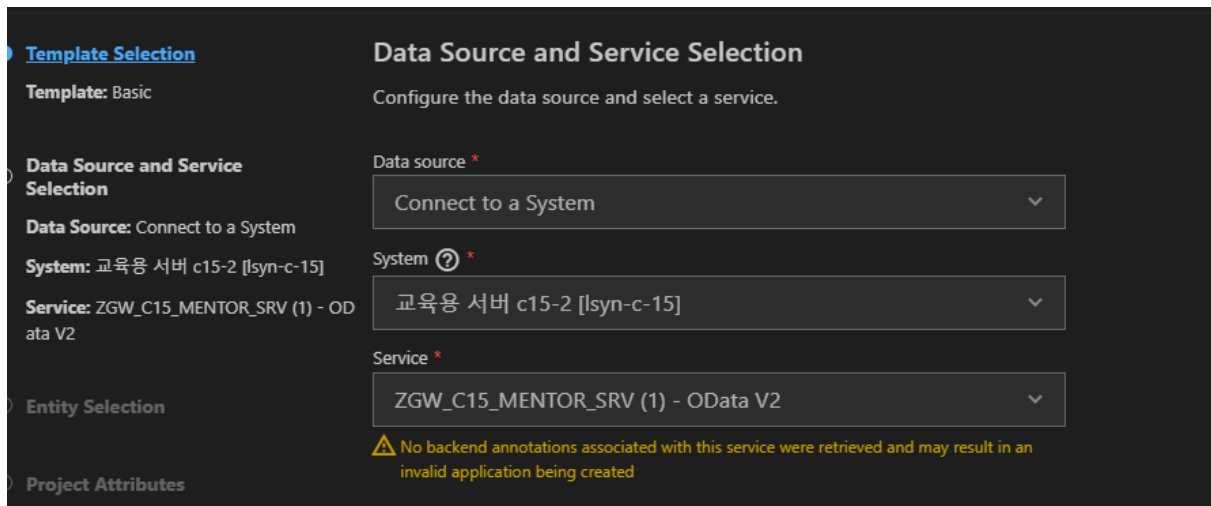
그리고 항상 ACTIVE 를 해줘라!!!! 안해주면 SEGW 에서 조회가 안됨

그리고 항상 ACTIVE 를 해줘라!!!! 안해주면 SEGW 에서 조회가 안됨

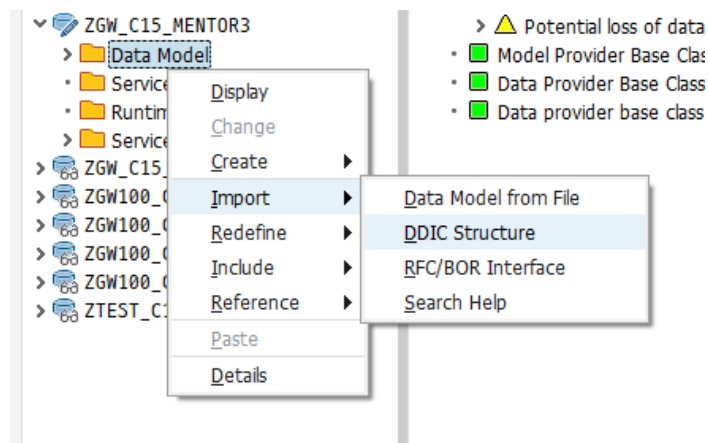
Step.2



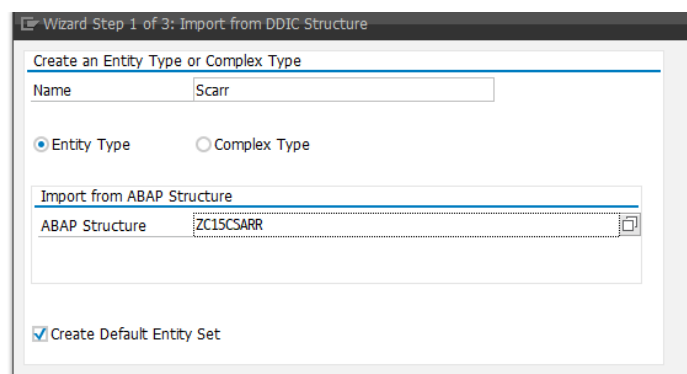
segw에서 project를 새로 생성한다.



Project의 이름은 우리가 UI5 에서 검색할 **odata service** 의 이름이 된다.

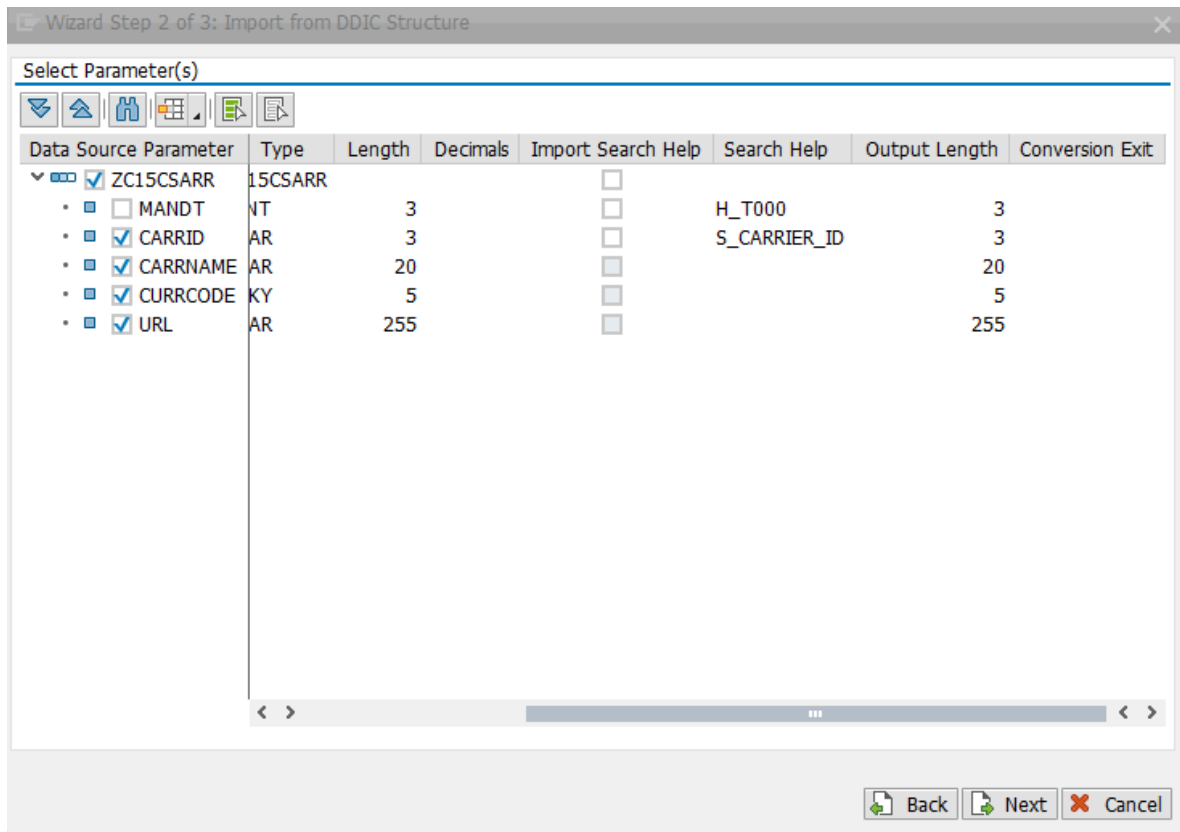


사용할 CDS VIEW 를 import → DDIC Sturcture (association을 할경우 view 2개를 모두 import)



name - entity 의 이름

ABAP Structure에 우리가 cds view 에서 @AbapCatalog.sqlViewName: 로 등록한 이름을 적는다.



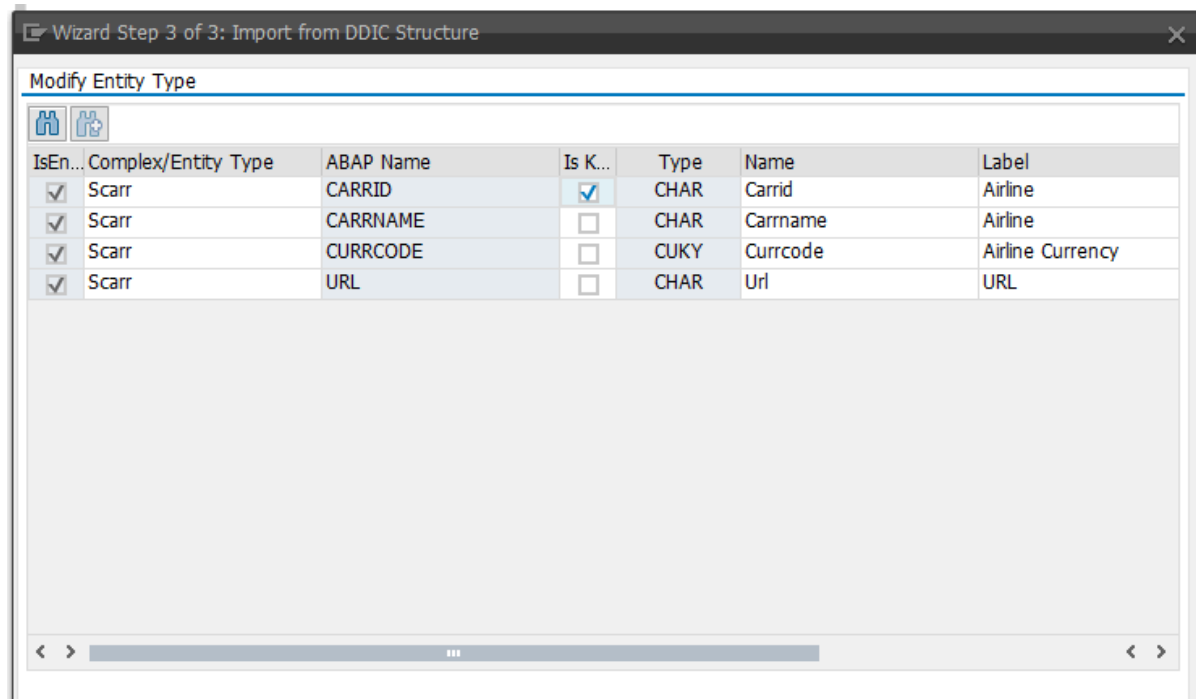
사용할 field 를 선택한다.

이때 Conversion Exit 를 꼭 확인해라.... 이 과정에서만 conversion exit 을 확인 가능하다!!!!

Fltime 처럼 view type 과 gw 에서 사용할 type 이 다른경우 나중에직접 바꿔줘야한다.

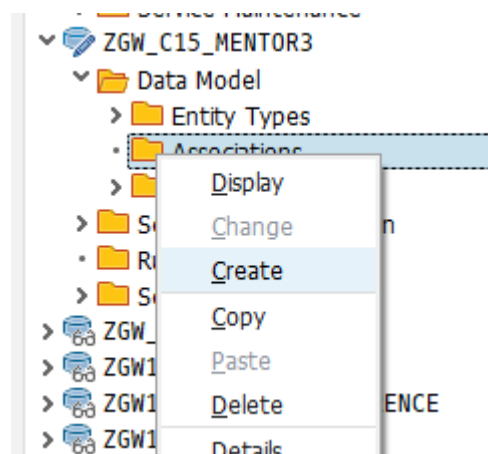
Properties						
Name	Is Key	Edm Core Type	Prec.	Scale	Max Ln...	Unit Property Name
Mandt	<input type="checkbox"/>	Edm.String	0	0	3	
Carrid	<input checked="" type="checkbox"/>	Edm.String	0	0	3	
Connid	<input checked="" type="checkbox"/>	Edm.String	0	0	4	
Countryfr	<input type="checkbox"/>	Edm.String	0	0	3	
Cityfrom	<input type="checkbox"/>	Edm.String	0	0	20	
Airpfrom	<input type="checkbox"/>	Edm.String	0	0	3	
Countryto	<input type="checkbox"/>	Edm.String	0	0	3	
Cityto	<input type="checkbox"/>	Edm.String	0	0	20	
Airpto	<input type="checkbox"/>	Edm.String	0	0	3	
Fltime	<input type="checkbox"/>	Edm.String	0	0	200	
Deptime	<input type="checkbox"/>	Edm.Time	0	0	0	
Arrtime	<input type="checkbox"/>	Edm.Time	0	0	0	
Distance	<input type="checkbox"/>	Edm.Decimal	9	4	0	Distid
Distid	<input type="checkbox"/>	Edm.String	0	0	3	
Fltype	<input type="checkbox"/>	Edm.String	0	0	1	
Period	<input type="checkbox"/>	Edm.Byte	0	0	0	

이런식으로 data model → entity type → properties 에서 수정을한다.



key field 를 선택한다. 이것은 원본 table 의 key field 를 따라서 체크한다.

Step.3



view 간의 accosiation 을 만든다.

Wizard Step 1 of 3: Create Association

Create Association

Association Name: Asso_Scarr_Sflight

Principal Entity		Dependent Entity	
Entity Type Name	Scarr	Entity Type Name	Sflight
Cardinality	1	Cardinality	0..n
<input checked="" type="checkbox"/> Create related Navigation Property		<input checked="" type="checkbox"/> Create related Navigation Property	
Navigation Property	toSflight	Navigation Property	toScarr

entity type 은 우리가 import 한 entity 이름

cardinality 는 view 에서 사용했던 관계(scarr 와 sflight 는 0:n 의 관계를 가진다.)

navigation property 는 우리가 odata service 에서 사용할 navigation property의 이름이다(ex - to_flight)

Wizard Step 2 of 3: Create Association

Referential Constraints

Principal Entity	Principal Key	Dependent Entity	Dependent Property
Scarr	Carrid	Sflight	Carrid

연결할 key field 를 선택한다.

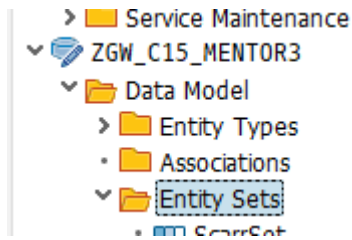
Wizard Step 3 of 3: Create Association

Create Association Set

Association Set Name: Asso_Scarr_SflightSet

Principal Entity		Dependent Entity	
Entity Set Name	Scarr001Set	Entity Set Name	SflightSet

entity set 의 이름을 적는다. 이때 자동으로 적혀 있는데 '001' 같은 값이 있는 경우가 있다
이럴때는



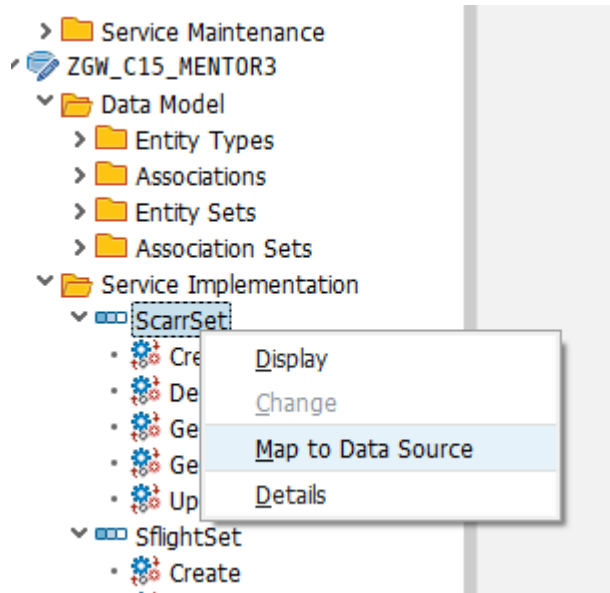
Entity Sets		
Name	Entity Type Name	Label
Scarr001Set	Scarr	
ScarrSet		
SflightSet	Sflight	

data model → entity sets 에 들어가서 다음을

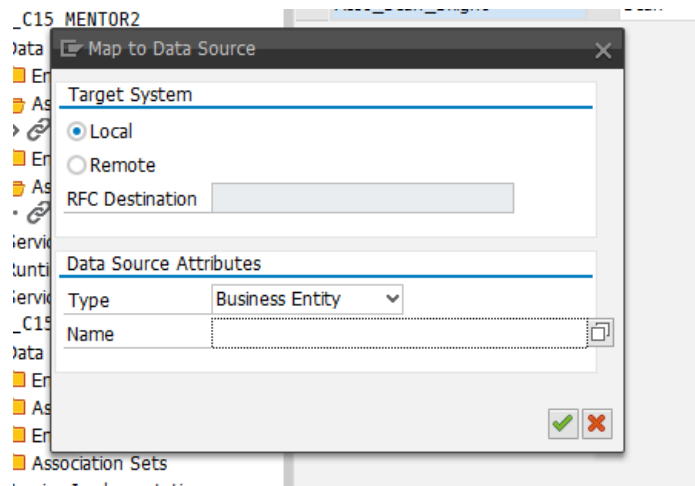
Entity Sets		
Name	Entity Type Name	Label
ScarrSet	Scarr	
SflightSet	Sflight	

이렇게 바꿔주면 된다.

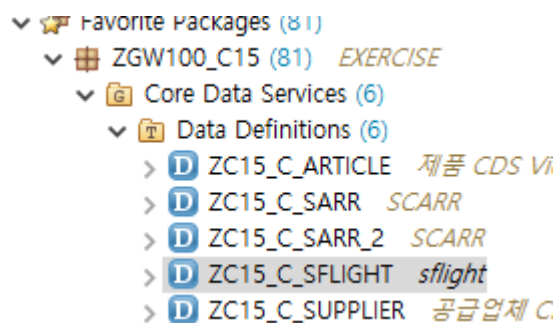
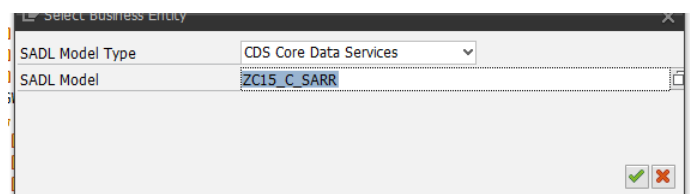
step.4



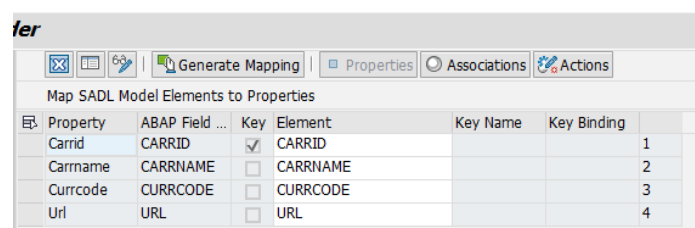
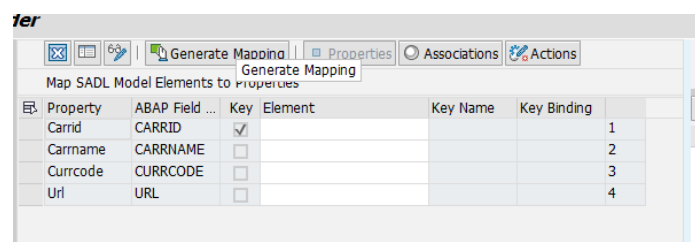
만들어진 서비스와 data source 간에 mapping을 진행한다.



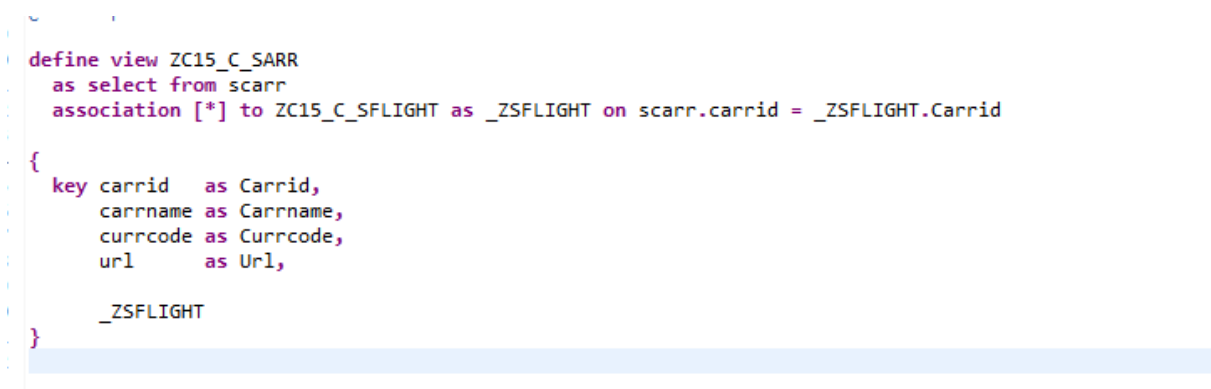
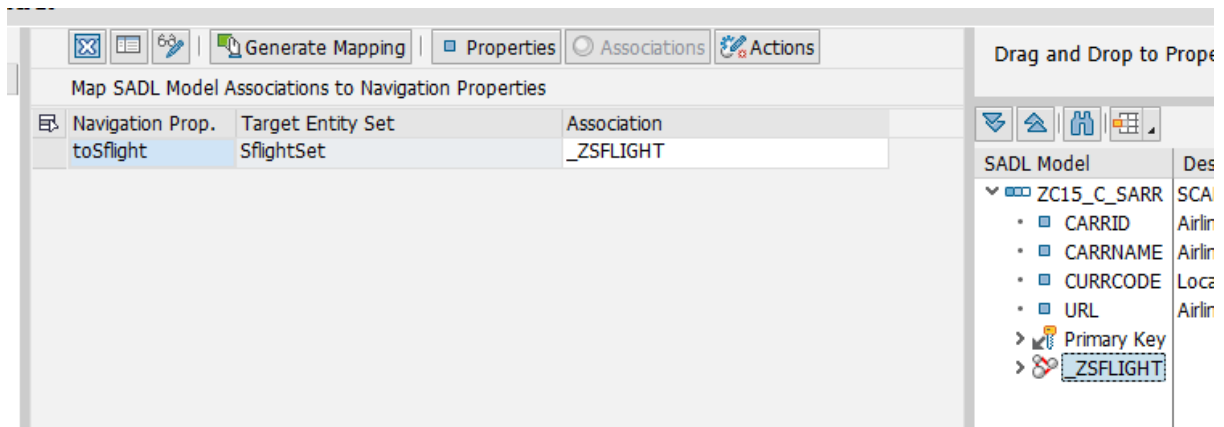
내임 옆의 조그만 종이 모양클릭



SADL model 에 우리가 이클립스에 만든 CDS View 의 data definition 을 입력한다.

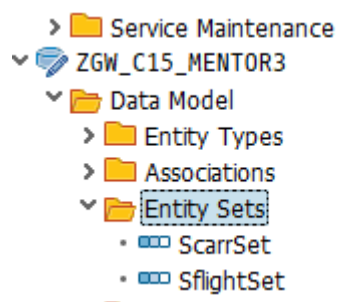


generator mapping 클릭

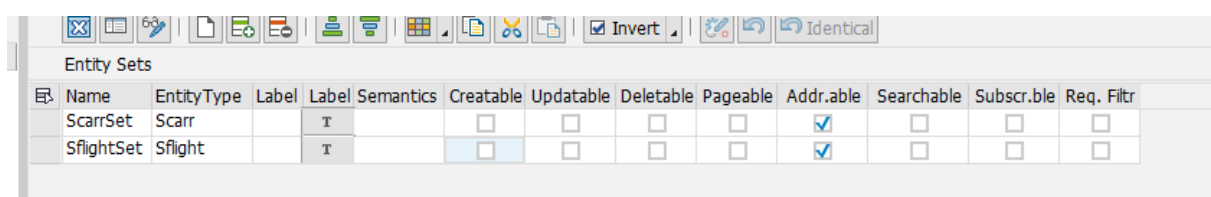


to navigation property 에 표시할 내용을 드래그 해서 옮긴다.

모든 entity 에 대하여 진행한다.



여기 폴더로 이동



Entity Sets													
Name	EntityType	Label	Label	Semantics	Creatable	Updatable	Deletable	Pageable	Addr.able	Searchable	Subscr.ble	Req. Filtr	
ScarrSet	Scarr		T		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
SflightSet	Sflight		T		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

CRUD 를 사용해야 함으로 다음과 같이 체크

그다음

generate 를 클릭하여 odata service 를 생성하라

참고로 이 과정 후에

- ▼ ScarrSet
 - Mapping
 - Create
 - Delete
 - GetEntity (Read)
 - GetEntitySet (Query)
 - Update

GetEntity(read) 와 GetEntitySet(Query) 는 자동으로 구현이 되어 진다.

왜?? cds view 에서 select 를 진행하여서

하지만 create, delete, update 는 구현이 되어 있지 않음으로 저 기능들은 따로 구현을 해줘야 한다.!!

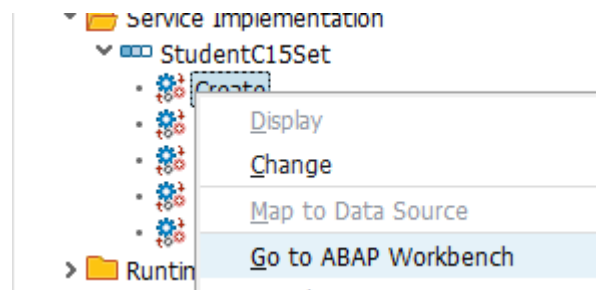
entity set 양쪽에 모두 구현을 해줘야함!!!! 한쪽만 구현하면 한쪽만 기능을 사용가능하다.

만약 조회만 사용할거면 따로 구현은 안해줘도됨, 한쪽만 삭제, 수정, 생성 기능이 필요하면 한 쪽에만 구현 해줘도 됨

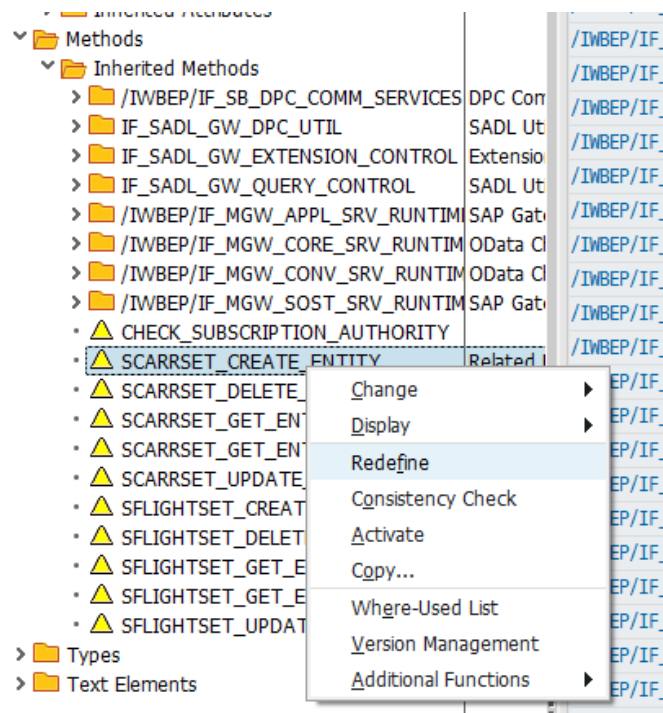
Step.5

create, delete, update 구현

이건 직접 DB Table 을 하나 만들어서 test 해봐라 여기 있는 sflight, spfli 같은 table 은 값을 수정 못하는 경우가 있음



이동



사용할 기능들을 redefine 해준다.

create

* 생성할 데이터를 가져온다.

```
io_data_provider→read_entry_data(  
    IMPORTING  
        es_data = er_entity  
    ).
```

* 사용자를 기준으로 데이터를 생성하기 직전에 현재시간을 기록

* 기록 하는 것을 추천 안해도 상관없음

* 해당하는 field를 data model → entity type → properties 에서 생성 하면됨

```
er_entity-erdat = sy-datum.
```

```
er_entity-erzet = sy-uzeit.
```

```
er_entity-ernam = sy-uname.
```

* db tab 에 데이터 저장

```
insert [CDS_VIEW 가 참고하는 DB TAB 이름] from er_entity. // CDS View 이름  
// Vie  
// DB
```

* 결과가 실패라면 EXCEPTION 을 발생시킨다.

```
CHECK sy-subrc <> 0.
```

```
DATA(LO_MSG_CONTAINER) = mo_context→get_message_container( ).
```

* 메세지 class 를 사용하여 호출 하는 방법 이렇게 안해도 됨.

```
lo_msg_container→add_message(  
    iv_msg_type      = 'E'           " Message Type  
    iv_msg_id        = 'ZGW100_C15_MSG' " Message Class  
    iv_msg_number     = '006'        " Message Number  
    ).
```

* 이거 사용해도됨.

```
mo_context→get_message_container( )→add_message_text_only(  
    EXPORTING  
        iv_msg_type = 'E'  
        iv_msg_text = '에러 메세지에 출력할 내용'  
    ).
```



```

RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
EXPORTING
    textid          = /iwbep/cx_mgw_busi_exception⇒business_error
    message_container = lo_msg_container
.

```

Delete

* entity type과 동일한 아باط 변수를 만들어서 키값을 받아 올 수 있다.

DATA ls_structe **TYPE** [CDS_VIEW 가 참고하는 DB TAB 이름].

```

io_tech_request_context⇒get_converted_keys(
    IMPORTING
        es_key_values = ls_structe          " Entity Key Values - converted
    ).

```

* **DATA** lt_keys **TYPE** /iwbep/t_mgw_tech_pairs.

* io_tech_request_context⇒get_keys(

* **RECEIVING**

* rt_keys = lt_keys " Technical name-value pair

*).

** DB TAB 의 레코드를 삭제하려면 [**DELETE FROM** dbtab **WHERE** 조건절] 문법을 사용한다.

** ITAB 의 데이터를 삭제할때는 [**DELETE** itab **WHERE** 조건절] 를 사용한다.

```

* DELETE FROM [CDS_VIEW 가 참고하는 DB TAB 이름]
    WHERE [연결 field] = ls_structe-[연결 field]

```

*

** 삭제 실패시는 오류처리를 하지 않는다. 데이터가 없어서 실패한 거기 때문에

* 검색하고 데이터가 있으면 삭제를 하도록한다.

```

SELECT SINGLE * FROM [CDS_VIEW 가 참고하는 DB TAB 이름] INTO ls_structe
    WHERE [연결 field] = ls_structe-[연결 field].

```

```

IF sy-subrc = 0.
" 반드시 ls_structe에 key 값이 기록되어 있어야한다.
DELETE FROM [CDS_VIEW 가 참고하는 DB TAB 이름]
WHERE [열결 field] = ls_structe-[열결 field].
ENDIF.

```

삭제시에는 메시지를 띄우지 않아도 된다. 데이터가 없어서 실패한 거기 때문에

Update

* 수정할 데이터를 가져온다.

```

io_data_provider→read_entry_data(
IMPORTING
es_data = er_entity
).

```

* db update : 절대 키필드의 값을 변경하지 않는다.

* UPDATE 시 절대로 키필드를 변경하면 안된다 기억

```

UPDATE [CDS_VIEW 가 참고하는 DB TAB 이름] SET name = er_entity-name
aedit = sy-datum " 수정일자를 현재일자로
aedit = sy-uzeit " 수정시간을 현재 시간으로
aedit = sy-uname " 수정자를 현재 사용자로
WHERE [열결 field] = ls_structe-[열결 field].
" 검색조건은 키필드로 구성한다.

```

* 업데이트 실패시

```

CHECK sy-subrc <> 0.

```

```

DATA(lo_msg_container) = mo_context→get_message_container().

```

```

mo_context→get_message_container()→add_message_text_only(
EXPORTING
iv_msg_type = 'E'
iv_msg_text = '에러 메시지에 출력할 내용'
).

```

```

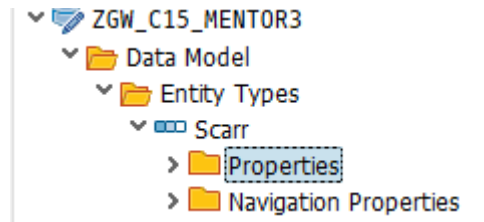
RAISE EXCEPTION TYPE /iwbep/cx_mgw_busi_exception
EXPORTING

```

textid = /iwbep/cx_mgw_busi_exception⇒business_error
message_container = lo_msg_container.

번외편

entity에 properties 추가



Data Model → properties 이동

Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit Prop.
Carrid	<input checked="" type="checkbox"/>	Edm.String	0	0	3	
Carrname	<input type="checkbox"/>	Edm.String	0	0	20	
Currcode	<input type="checkbox"/>	Edm.String	0	0	5	
Url	<input type="checkbox"/>	Edm.String	0	0	255	

위에 초록색 + 가 있는 테이블 모양의 아이콘을 클릭

Name	Key	Edm Core Type	Prec.	Scale	Max...	Unit Prop.
	<input type="checkbox"/>					
Carrid	<input checked="" type="checkbox"/>	Edm.String	0	0	3	
Carrname	<input type="checkbox"/>	Edm.String	0	0	20	
Currcode	<input type="checkbox"/>	Edm.String	0	0	5	
Url	<input type="checkbox"/>	Edm.String	0	0	255	

추가됨

oData servie 에서 조회할 어려운 옵션 Query 문 예시

gpt 로 만들었는데 작동 하는 것 확인함

✅ 1. 상위 6개만 조회 (\$stop)

/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?\$top=6

?\$ 로 시작을 한다.

✅ 2. 특정 항공사(CARRID) 필터링 (\$filter)

예: CARRID 가 "AA"인 항공사 조회

```
0 define view ZC15_C_SARR
1   as select from scarr
2   association [*] to ZC15_C_SFLIGHT as
3
4   {
5     key carrid as Carrid,
6     carrname as Carrname,
7     currcode as Currcode,
8     url as Url,
9
10    _ZSFLIGHT
11  }
```

/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?\$filter=Carrid eq 'AA'

\$filter=CARRID 에서 field 명은 우리가 CDS View 의 Field 명으로 입력한 내용

✅ 3. 항공사 이름(SCARRNAME) 기준으로 정렬 (\$orderby)

예: SCARRNAME 기준 오름차순 정렬

/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?\$orderby=Carrname asc

✅ 4. 특정 조건 + 정렬 + 상위 6개 조회 (조합 가능)

예: CARRID 가 "AA"이고, SCARRNAME 기준 오름차순 정렬 후 상위 6개 조회

```
/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?$filter=CARRID eq 'AA'&$orderby=Carrname asc&$top=6
```

조건들의 연결은 & 로 이루어진다.

✓ 5. 여러 개의 필터 조건 사용 (and , or)

예: **CARRID** 가 "AA"이거나 "LH"인 항공사 조회

```
/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?$filter=Carrid eq 'AA' or Carrid eq 'LH'
```

예: **CARRID** 가 "AA"이고, **SCARRNAME** 이 "American Airlines"인 경우만 조회

```
/sap/opu/odata/SAP/ZGW_C15_MENTOR2_SRV/ScarrSet?$filter=Carrid eq 'AA' and Carrname eq 'American Airlines'
```

\$expand=Orders 이거도 기억

근데 저거 안써도 노상관 navigation property 를 부를때 미리 불러서 서버에서 안읽고 가져오냐,

이차이여서. 코드상 차이는 없음

OData의 **\$expand** 를 사용하면 **네비게이션 프로퍼티** 데이터를 함께 가져올 수 있습니다.

아래 예제에서 **\$expand** 를 어떻게 활용하는지 확인해 보세요.

◆ 예제 1: **bindElement()** 에서 **\$expand** 사용

javascript

복사편집

```
var sPath = "/Supplier(guid'XYZ')?$expand=Orders";  
this.getView().bindElement({
```

```

path: sPath,
parameters: {
  expand: "Orders" // 네비게이션 프로퍼티 'Orders'도 함께 가져옴
}
});

```

👉 위 코드에서는 **Supplier** 엔티티를 바인딩하면서, **Orders** 라는 네비게이션 프로퍼티도 같이 가져옵니다.

✅ 비교 정리

경우	READ 요청 발생?	네비게이션 프로퍼티 (Orders) 가져옴?
<code>bindElement("/Supplier(guid'xxx')")</code>	✅ 발생함	❌ 안 가져옴
<code>bindElement("/Supplier(guid'xxx')", { expand: "Orders" })</code>	✅ 발생함	✅ 같이 가져옴
<code>bindElement("/Supplier(guid'xxx')")</code> 후 <code>oModel.read("/Supplier(guid'xxx')/Orders")</code>	✅ 두 번 발생	✅ 가져옴 (두 번째 요청에서)

📌 결론:

- ✅ **\$expand** 를 써도 **READ** 요청은 발생함.
- ✅ 하지만 추가적인 **READ** 요청 없이 네비게이션 프로퍼티 데이터를 한 번에 가져올 수 있음.
- ❌ **\$expand** 를 안 쓰면, 네비게이션 프로퍼티 데이터는 자동으로 안 가져오고 추가 요청이 필요