



왕초보 세션 1



질문

⚡ **알아가봅시다 !**

- Data Elements, Data Type, Data Object
- 변수, Type 선언(Declare)과 Data 정의(Define).

- Data element vs Domain의 차이
- Value , Structure , Internal table , DB Table

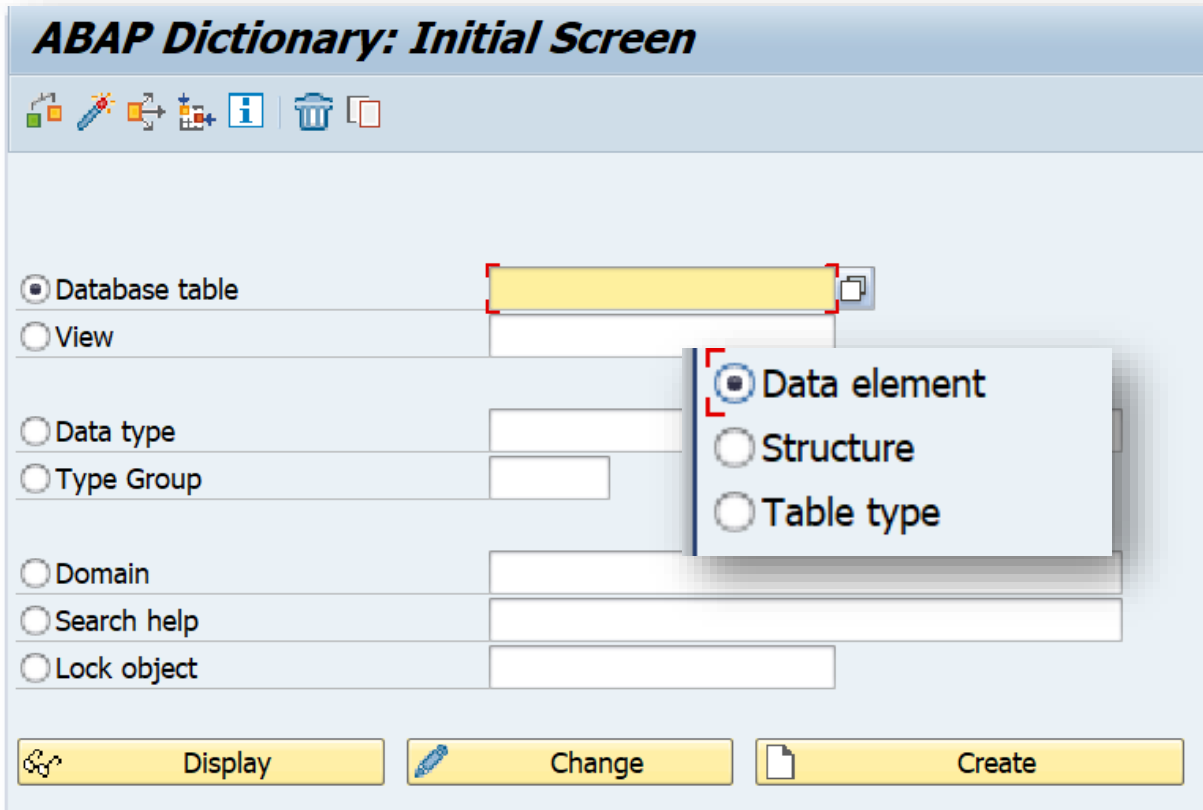
- 문제를 읽고 적절한 키워드와 코드작성하는 방법
- 코드 작성 시 Select endselect 방식과 array fetch 방식 차이랑 돌아가는 과정을 코드리뷰

1. Data Type 선언 & Data Object 정의 (Declare) (Define)

2. ABAP Dictionary

3. 퀴즈-실습 ABAP Quiz 16.

ABAP Dictionary 이거 왜 쓸까요?



Purpose

You use the ABAP Dictionary to create and manage data definitions (metadata). The ABAP Dictionary permits a central description of all the data used in the system without redundancies. New or modified information is automatically provided for all the system components. This ensures data integrity, data consistency and data security.

The ABAP Dictionary supports the definition of user-defined types (data elements, structures and table types). You can create the corresponding objects (tables or views) in the underlying relational database using these data definitions. The ABAP Dictionary describes the logical structure of the objects used in application development and shows how they are mapped to the underlying relational database in tables or views.

The ABAP Dictionary also provides standard functions for editing fields on the screen, for example for assigning input help to a screen field.

System에서 사용되는 모든 Types 들에 대한

✓ 조회 -> Display

✓ 수정 -> Change

✓ 생성 -> Create

System에서 사용되는 모든 “Types”들에 대한?

Data Types

In ABAP Dictionary, global data types can be defined that are visible for all repository objects in the current AS ABAP.

The possible data types in ABAP Dictionary are:

- **Data Elements** for elementary data types and reference types
 - **Structures** that can comprise any other number of data types.
 - **Table Types** with any row type
-
- Data Elements – 기초 데이터 타입 & 참조 타입
 - Structures – 다른 여러 데이터타입을 조합하는
 - Table Type – 행 타입을 갖는

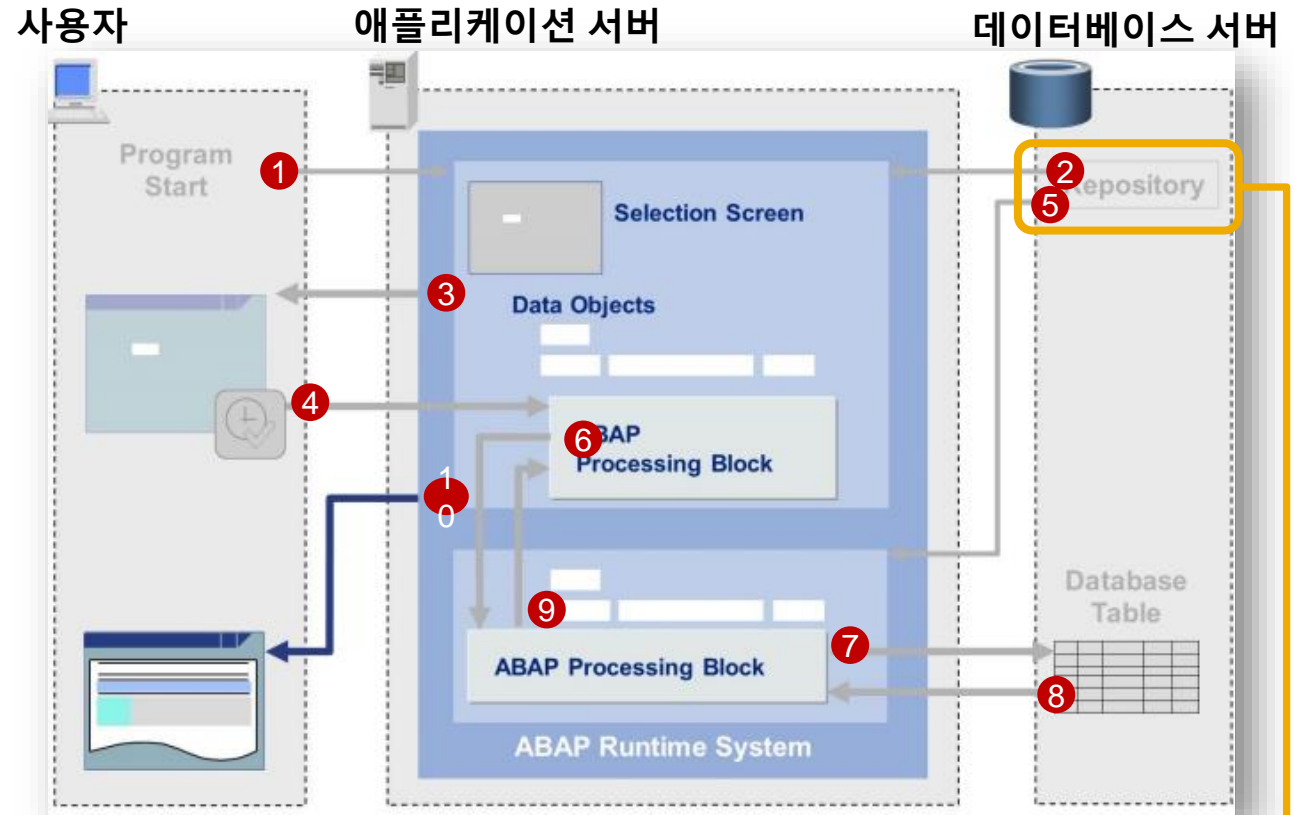
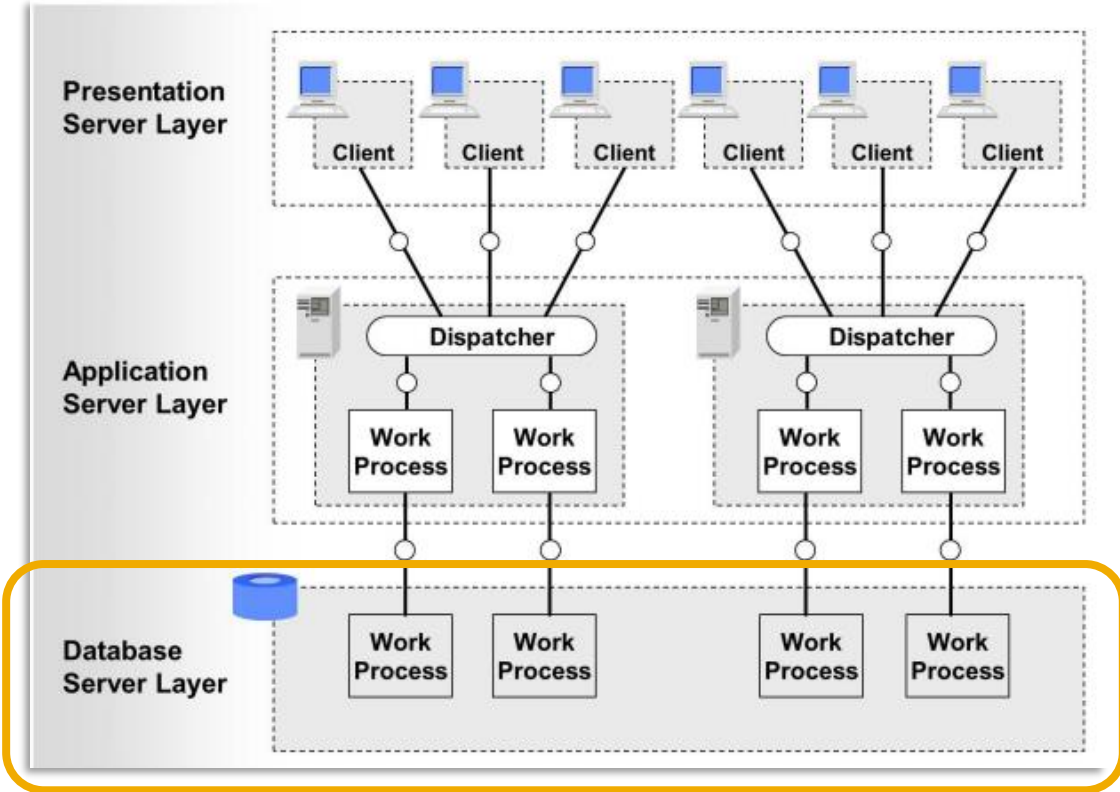
ABAP Dictionary에서는

- 현재 ABAP **Application Server**내에있는
- 모든 **Repository objects**가 사용(접근)할 수 있는
- **Global Data Types**를 정의.

[암기 포인트]

- ✓ ABAP Dictionary에서는 오직 **Types** 만 선언 할 수 있다.

Define Global Data Types & Visible All Repository objects



✓ ABAP Dictionary에 선언되어있는 Global Data Type들은 데이터베이스 서버의 레포지토리에 저장되어있어, 같은 DB서버를 사용하는 사용자들은 다른 사용자가 만들어낸 타입을 공유하여 접근, 조회, 수정, 삭제 할 수 있다.

목차

⚡ 함께 실습해요!

1 변수의 종류와 선언 방법 ⚡

2 TABLE의 종류
Transparent / DB / Internal Table

3 SELECT문 ⚡

4 Internal Table Handling
LOOP문 / READ TABLE / INSERT / APPEND

목차

1 변수는 왜 써야 하는가?
- 값(데이터)를 담으려고

2 테이블로 무엇을 할 것인가?
- 데이터에 접근하고 컨트롤

3 SELECT문은 왜 사용하는가?
- 내가 원하는 데이터를 가져오기 위해

4 Internal Table Handling
LOOP문 / READ TABLE

1-1. 변수의 종류

실제로 데이터를 담을 수 있는 Data Object
Naming Rule에 따라 지어짐

DATA gs_sflight **TYPE** sflight.

ABAP Dictionary에 선언되어있는 Global Data Type
또는
프로그램 내에서 선언한 Local Data Type

1-1. 변수의 종류

Data Element

A data element defines an elementary data type or a reference type and the technical type attributes. It describes the semantic meaning of a (database) field – for example, the header for table displays or documentation text. Properties such as data type, length, and so on, are defined directly in the data element or defined in a domain.

- Semantic, 의미적인 이란 뭘까?

Use

You use a data element to define a data type that can be typically used as the type of a table field or a structure component.

- 기초적인 데이터 타입 & 참조타입을 정의.
- 테크니컬 속성 – 길이, 문자형 & 정수형
- 필드의 시멘틱 의미 -

1-1. 변수의 종류

Structure

Use

You use a structure (structured type) to **comprise components** (fields). Types are defined for the components. A component can refer to an **elementary type** (via a data element or by directly specifying the data type and length in the structure definition), another **structure** or a **table type**. A structure can be nested to any depth.

Features

Structures are used to define the data at the interface of module pools and screens and to define the parameter types of function modules.

The central definition of structures that are used more than once makes it possible for them to be changed centrally. The active ABAP Dictionary then makes this change wherever required. The system automatically adjusts ABAP programs or screen templates that use a structure when the structure is changed. This ensures **the greatest possible consistency** of the data definition, also for complex programs.

- 컴포넌트들을 조합하기 위해 사용.
- Elementary type, structure type, table type 또한 컴포넌트가 될 수 있음.

1-1. 변수의 종류

Table Type

ABAP Dictionary

Use

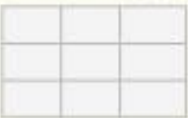
You use a table type to describe the structure and functional attributes of an internal table in ABAP. In ABAP programs you can reference a table type TTYP defined in the ABAP Dictionary with the command `DATA <inttab> TYPE TTYP`. You create an internal table <inttab> in the program with the attributes defined for TTYP in the ABAP Dictionary.

Structure



Structure Type

Table Type



Type for Internal Table

- Internal Table의 기능적 요소와 구조를 기술
- 프로그램에서 테이블 타입을 참조할 수 있음.

1-1. 변수의 종류

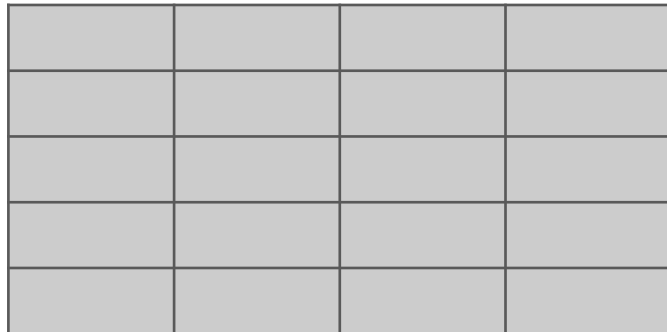


Data Element

Naming Rule



Structure



Internal Table



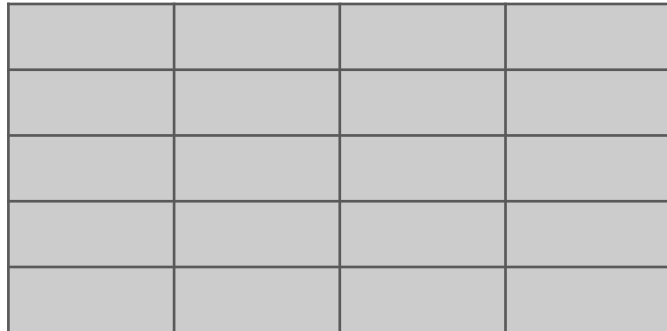
1-1. 변수의 종류



Data Element



Structure



Internal Table

Naming Rule

gv_name
Global Variable

gs_name
Global Structure

gt_name
Global Table

변수의 종류 (ABAP Dictionary Type)

데이터 타입(틀,설계도)로 데이터 오브젝트(그릇,건물)을 찍어내야
비로소 그(그릇,건물)안에 벨류(계란,가전기기)을 담을 수 있다.

Type:

그릇을 만들어 내는 틀

→ 변수(variable)의 기술적 속성 등을
정의하기 위해 사용.

Data Object(Data Variable)

: 그릇

→ Data를 담을 수 있는 것

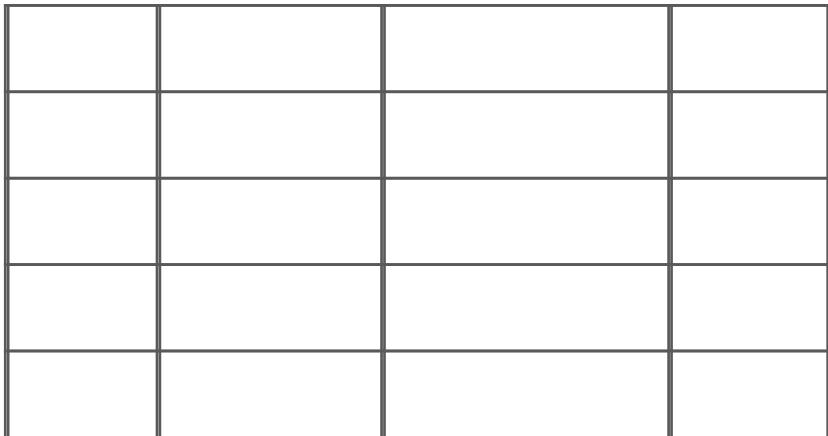


Data type은 Data variable과 독립적으로 선언

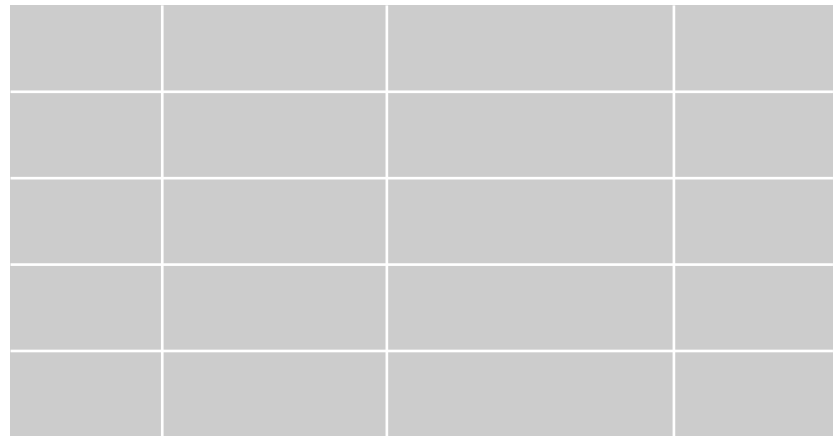
Data variable은 Data type을 참조하여 선언.

1-1. 변수의 종류

Data Type - 설계도



Data Object - 건물



갓 정의된 데이터 오브젝트에는 텅 비어있다.

1-1. 변수의 종류

Data Type - 설계도

--

--	--	--	--

Data Object – 건물
이 안에 값을 채움.

2020.10.01

AA	0017	2020.10.01	P
----	------	------------	---

AA	0017	2020.10.01	P
AA	0017	2020.10.01	B
LH	0400	2020.10.01	P
AA	0017	2020.10.01	P
DL	1984	2020.10.01	B

데이터 오브젝트에 값을 넣어줘야 의미를 지닌다.

1-2. 변수의 선언방법

Data Element



```
DATA : name1 TYPE s_carr_id.
```

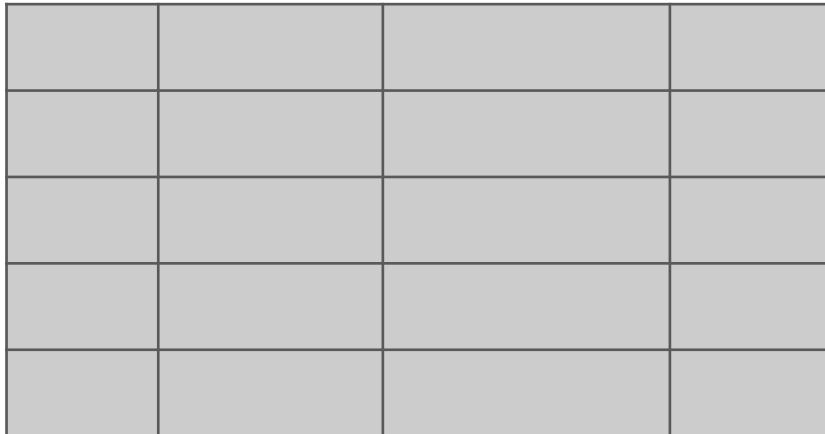
Structure



```
DATA : name2 TYPE bc400_s_flight.
```

```
DATA : name4 TYPE scarr
```

Internal Table



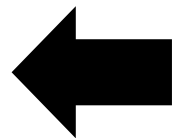
```
DATA : name3 TYPE bc400_t_flights.
```

1-2. 변수의 선언방법



Element

```
DATA : gv_percentage TYPE P LENGTH 3 DECIMALS 2 .  
DATA : gv_num1 TYPE i . DATA : name2 TYPE bc400_s_flight.  
DATA : gv_city TYPE c LENGTH 14.  
-----  
DATA : gv_carrid TYPE s_carr_id .
```



Type
Length

1-2. 변수의 선언방법



Structure

Global DATA : gs_flight TYPE bc400_s_flight.

type : char
length : 3

type : numc
length : 4

type : dats
length : 8

type : int4
length : 10

type : int4
length : 10

type : dec
length : d
decimal : 2

carrid

connid

fldate

seatsmax

seatsocc

percentage

gs_flight



field

1-2. 변수의 선언방법

Table

```
DATA : gt_itab TYPE < Table Type > .
```

```

                STANDARD
DATA : gt_itab TYPE SORTED   TABLE OF bc400_s_flight
                HASHED
                WITH NON-UNIQUE KEY    ... .
                UNIQUE
    
```

```
DATA : gt_itab TYPE TABLE OF < Structure Type > .
```




Type field Standard/sorted/hashed
 Length Unique / non-unique

1-2. 변수의 선언방법

Local Structure Type 선언 – 프로그램 단위에서만 사용

```

TYPES : BEGIN OF ts_type,
    carrid    TYPE sbook-carrid,
    connid    TYPE sbook-connid,
    fldate    TYPE sbook-fldate,
    bookid    TYPE sbook-bookid,
    customid  TYPE sbook-customid,
    custtype  TYPE sbook-custtype,
    carrname  TYPE scarr-carrname,
    name      TYPE scustom-name,
END OF ts_type.

```

type : char length : 3	type: numc length : 4	type : dats length : 8	type: numc length : 8	type :numc length : 8	type : char length : 1	type : char length : 20	type : char length : 25
carrid	connid	fldate	bookid	customid	custtype	carrname	name

1-2. 변수의 선언방법

- 실습

1. (1) 숫자(int) 타입의 데이터 변수 gv_num을 선언
(2) 문자(character) 타입의 데이터 변수 gv_string을 선언
2. BUKRS라는 Data Element을 참조하는 변수를 선언 (2가지 방법)
// **hint** EKPO 테이블의 필드입니다.

~~+) ZCL3TABLE 이 테이블의 필드 ID 타입을 따르는 변수를 선언해보세요.~~

1-2. 변수의 선언방법

- 실습

1. (1) 숫자(int) 타입의 데이터 변수 gv_num을 선언
(2) 문자(character) 타입의 데이터 변수 gv_string을 선언
2. BUKRS라는 Data Element을 참조하는 변수를 선언 (2가지 방법)
// **hint** EKPO 테이블의 필드입니다.
3. 아래 필드를 가진 STRUCTURE를 로컬에 생성 // **hint** B_ _ _ _ OF ~

Field	Data Element
BUKRS	BUKRS
BELNR	BELNR_D

4. 3번의 STRUCTURE를 참조하는 INTERNAL TABLE 선언

1-2. 변수의 선언방법

- 실습

1. (1) 숫자(int) 타입의 데이터 변수 gv_num을 선언
(2) 문자(character) 타입의 데이터 변수 gv_string을 선언
2. BUKRS라는 Data Element을 참조하는 변수를 선언 (2가지 방법)
// **hint** EKPO 테이블의 필드입니다.
3. 아래 필드를 가진 STRUCTURE를 로컬에 생성 // **hint** B_ _ _ _ OF ~

Field	Data Element
BUKRS	BUKRS
BELNR	BELNR_D
4. 3번의 STRUCTURE를 참조하는 INTERNAL TABLE 선언
5. 3번 필드를 가진 Transparent 테이블을 패키지에 생성 // ztable_
6. 5번에서 생성한 STRUCTURE를 참조하는 INTERNAL TABLE 선언

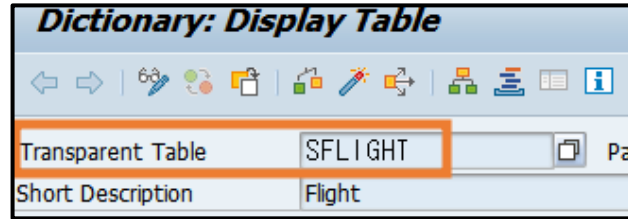
1-2. 변수의 선언방법

- 실습

7. 파라미터 변수 두 개(숫자 i) 값을 받아 더한 결과값을 출력하세요.
 - 1) 파라미터 변수 두 개 선언 pa_num1 , pa_num2
 - 2) 결과값을 담을 변수 선언 gv_result
 - 3) 파라미터 변수 두 개를 결과값에 반영할 로직 작성

2-1. Table의 종류

Transparent Table



Database Table 생성 목적
DB에 들어있는 데이터를 볼 수 있음
Structure Type

Database Table

SFLIGHT DB

MANDT	CARRID	CONNID	FLDATE	PLANETYPE	SEATSMAX
100	AA	0017	2020.11.02	747-400	385
100	AA	0017	2021.09.03	747-400	385
100	AA	0064	2021.05.15	A340-600	330
100	DL	1699	2021.03.08	767-200	260
100	DL	1984	2021.01.03	A380-800	475
100	KA	0064	2020.11.04	A340-600	330

Database에 존재하는 Table
Transparent Table과 1:1관계로
동일한 table명, 필드명을 가짐

Internal Table

gt_sflight

MANDT	CARRID	CONNID	FLDATE	PLANETYPE	SEATSMAX
100	AA	0017	2020.11.02	747-400	385
100	AA	0017	2021.09.03	747-400	385
100	AA	0064	2021.05.15	A340-600	330
100	DL	1699	2021.03.08	767-200	260
100	DL	1984	2021.01.03	A380-800	475
100	KA	0064	2020.11.04	A340-600	330

Program 내에서 Handling을 위해
DB Table에서 가져온 여러 개의 데이터를 담는 공간
Program 내에서 선언

**Transparent Table이라는 설계도로 DB Table을 만듭니다. (1:1관계)
애플리케이션 서버에 Internal Table**

2-1. Table의 종류 – Transparent Table

▼ Transparent Table and Structures

Transparent Table

- : 테이블 정의서(설명서)이며, ABAP Dictionary에서 정의
- : DB에 테이블을 생성하는 것이 목적
- : Transparent Table을 activate 하면 DB에 테이블이 자동 생성됨
- : Transparent Table과 DB Table은 1:1 관계

Transparent Table vs Structures

- 유사한 구조
 - : 모두 Component와 Component Type을 가짐
 - ABAP Program에서 Transparent table이 structure 타입 역할을 함

```
* Structure type으로 변수 선언.  
DATA wa1 TYPE s_struct.  
  
* Transparent table로 변수 선언.  
DATA wa2 TYPE spfli.  
  
* wa1, wa2 모두 Structure variable임.
```

- 차이점
 - : Structure의 경우 Component에 mandt가 없고 key field를 가지지 않음

[HOW TO - Transparent table]

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/17aeb44a-b97d-4094-b18e-4cccc262f89f/ABAP_Dictionary-Transparent_Table.pdf

3. SELECT 문

```
SELECT * FROM sbook
INTO CORRESPONDING FIELDS OF TABLE gt_tab
WHERE carrid IN s_car AND
      connid IN s_con AND
      fldate IN s_fld.
```

**Database Table에서
특정 열(필드)
특정 행(레코드)를 읽어와서**

**Internal Table에
담는다.**



sbook

Table: SBOOK
Displayed Fields: 22 of 24 Fixed Columns: 5

	MANDT	CARRID	CONNID	FLDATE	BOOKID	CUSTOMID	CUSTTYPE	SMOKER	L
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000061	00002532	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000065	00002448	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000069	00000279	B		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000070	00003307	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000075	00000096	B		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000093	00003817	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000139	00002404	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000144	00002633	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000203	00000138	B		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000209	00002247	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000275	00003424	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000307	00001501	P		
<input type="checkbox"/>	100	AA	0017	2020.10.01	00000350	00000911	P		

gt_tab

carrid	connid	fldate	bookid	customid	custtype	carrname	name

3. SELECT 문

Q1

Transparent / Database / Internal Table

```
DATA gs_sflight TYPE sflight.
```

Q2

여기 들어갈 키워드는?

```
DATA gt_sflight [ ] OF sflight.
```

Q3

Transparent / Database / Internal Table

```
SELECT carrid connid fldate  
FROM SFLIGHT  
INTO CORRESPONDING FIELD [ ]  
WHERE carrid = 'AA'.
```

Q4

여기 들어갈 키워드는?

Q5

Transparent / Database / Internal Table 의 FILED

3. SELECT 문

Q1

Transparent / Database / Internal Table

```
DATA gs_sflight TYPE sflight.
```

Q2

여기 들어갈 키워드는?

TYPE TABLE OF

```
DATA gt_sflight OF sflight.
```

Q3

Transparent / Database / Internal Table

```
SELECT carrid connid fldate  
FROM SFLIGHT  
INTO CORRESPONDING FIELD  
WHERE carrid = 'AA'.
```

Q4

여기 들어갈 키워드는?

TABLE gt_sflight

Q5

Transparent / Database / Internal Table 의 FILED

3. SELECT 문



에서 데이터를 가져오기 위해 사용하는 구문

어떤 DB Table에서 어떤 FIELD를 가져올 거야?

가져올 데이터의 조건은?

어디에 담을 거야?

3. SELECT 문

 DB 에서 데이터를 가져오기 위해 사용하는 구문

어떤 DB Table에서 어떤 FIELD를 가져올 거야?

FROM

SELECT


가져올 데이터의 조건은?

WHERE

어디에 담을 거야?

INTO

SELECT

 DB 에서 가져오고 싶은 FIELD


FROM

데이터를 가져올  DB TABLE

INTO

가져온 DATA를 담을 공간

WHERE

 DB 에서 데이터를 가져올 때의 조건

3. SELECT 문

SELECT
FROM
INTO
WHERE

1 개
RECORD

SELECT SINGLE
FROM
INTO
WHERE

 모든 KEY FIELD에 대하여
조건을 설정해야 함

여러 개
RECORD

SELECT
FROM
INTO TABLE
WHERE

여러 개의 ROW가 존재하기 때문에
TABLE에 데이터를 담아야 함

3. SELECT 문

SELECT
FROM
INTO
WHERE

1 개
RECORD

SELECT SINGLE
FROM
INTO CORRESPONDING FIELDS OF
WHERE

DB 모든 KEY FIELD에 대하여
조건을 설정해야 함

여러 개
RECORD

SELECT SINGLE
FROM
INTO CORRESPONDING FIELDS OF TABLE
WHERE

여러 개의 ROW가 존재하기 때문에
TABLE에 데이터를 담아야 함

DB 테이블과 STRUCTURE/TABLE의 FIELD 구성과 순서가 다른 경우가 많기 때문에

CORRESPONDING FIELDS OF 키워드를 항상 써주자

3. SELECT 문

- 실습

1. Selection Screen에서 받을 파라미터 변수 선언하기
2. 참조할 테이블의 STRUCTURE와 INTERNAL TABLE 선언하기
3. SELECT문을 통해 내가 찾을 정보를 특정하기
4. 출력 구문을 통해 내 이름을 출력하기
5. 나이도 출력해보고 내 나이를 맞추면 성공!

3. SELECT 문

- 실습

1. Selection Screen에서 받을 파라미터 변수 선언하기
2. 참조할 테이블의 STRUCTURE와 INTERNAL TABLE 선언하기
3. SELECT문을 통해 내가 찾을 정보를 특정하기
4. 출력 구문을 통해 내 이름을 출력하기
5. 나이도 출력해보고 내 나이를 맞추면 성공!

☺ 심화문제) INNER JOIN

3. SELECT 문

- 실습 심화

1. 다음 세션에 진행 예정입니다 -!

4. Internal Table Handling

LOOP문 / READ TABLE / INSERT / APPEND

LOOP AT

Internal Table을 가공, 조회하기 위해 행을 순서대로 스트럭처에 복사해 내부 실행문을 반복하여 실행한다.

LOOP AT [ITAB] INTO [WA]

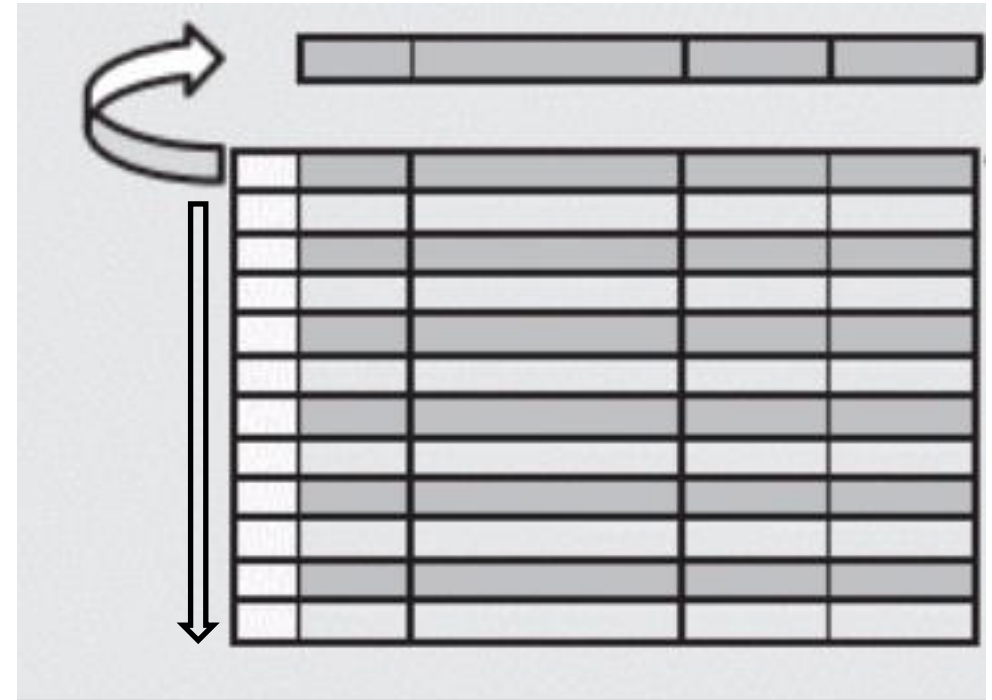
[WHERE 조건]

[FROM start_index TO end_index].

[내부 실행문]

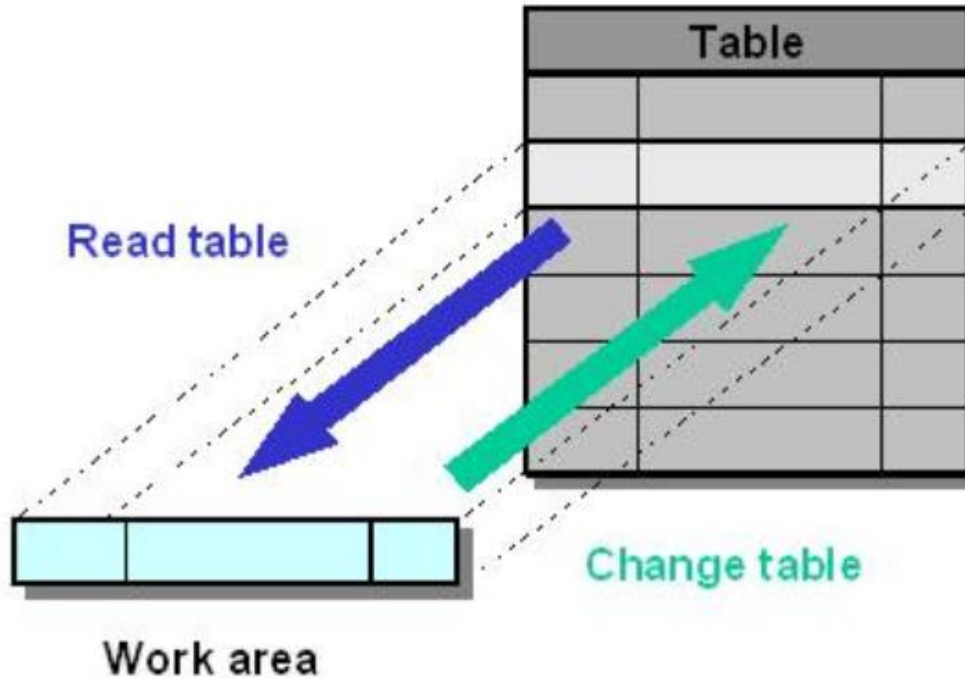
END LOOP.

INTO 절 - [ITAB]의 행을 차례대로 저장 할 스트럭처를 적는다.



Read table

Internal Table의 특정 레코드를 읽어올 때 사용하는 문법



READ TABLE [ITAB] INTO [WA] WITH KEY [필드명] = [값]

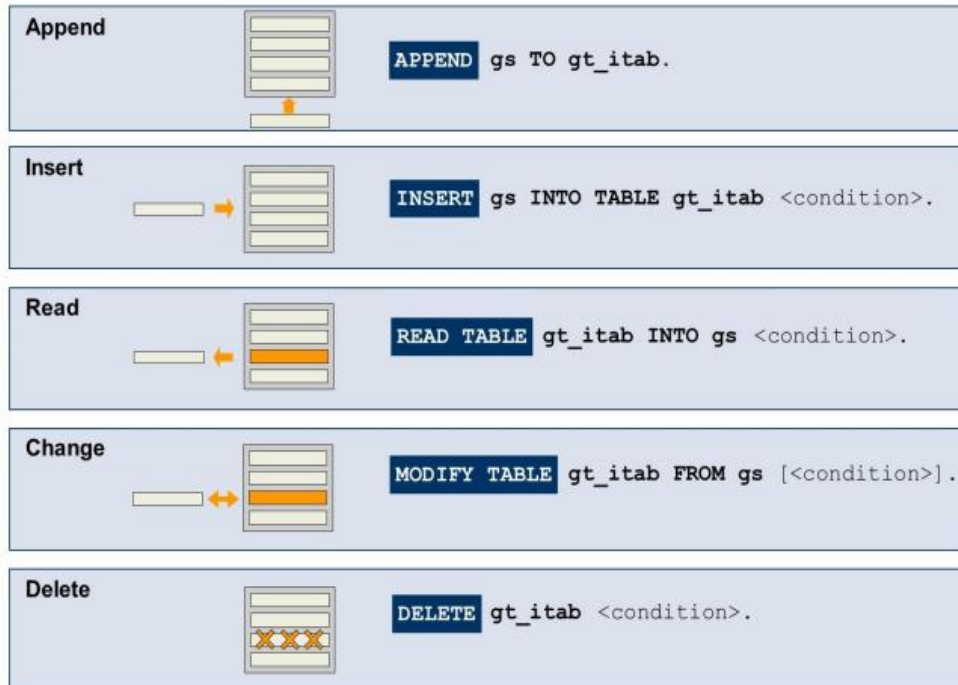
READ TABLE [ITAB] INTO [WA] INDEX = [N]

Internal Table 명령어

TAW10

Figure 184: Single Records Access (Overview)

Unit 11 Lesson 2

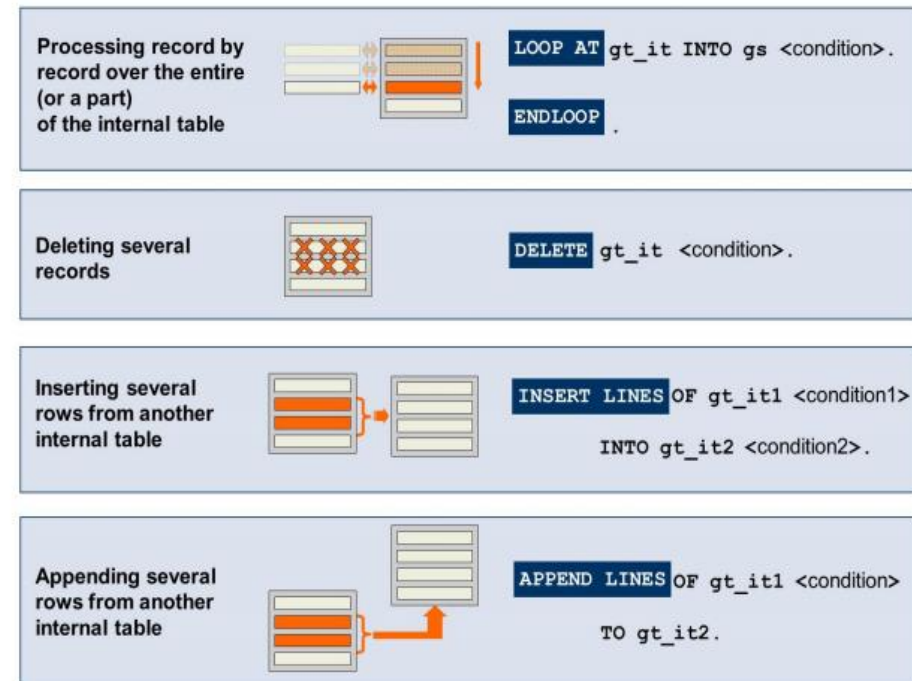


This material is for the exclusive use of SAP Trainer

TAW10

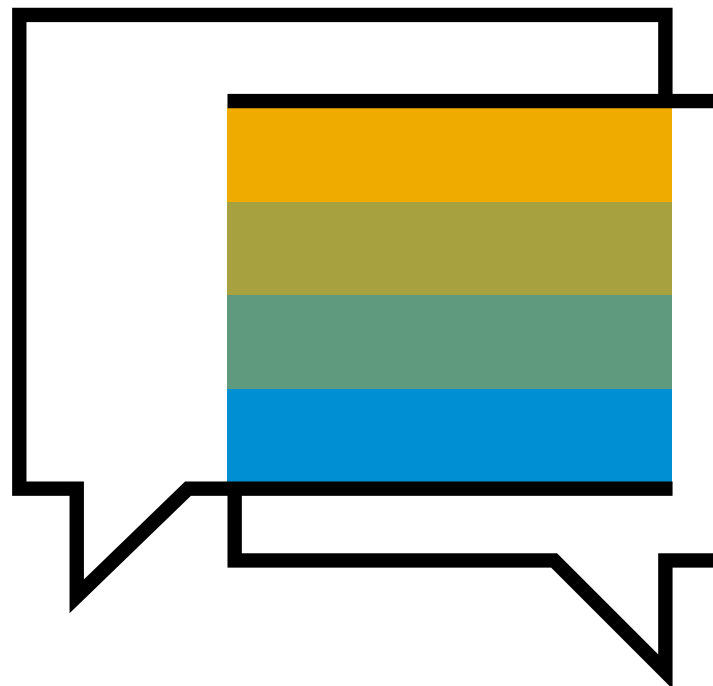
Figure 185: Processing of Sets of Records (Overview)

Unit 11 Lesson 2



This material is for the exclusive use of SAP Trainer

Q & A



TABLES

문법

TABLES : `table_wa`.

효과

This statement is not allowed in classes and declares a data object `table_wa` as a table work area whose data type is taken from the identically named structured data type `table_wa` in ABAP Dictionary. `table_wa` must be defined as a flat structure in ABAP Dictionary.

This means either database tables or classic views can be specified for `table_wa`.

Table work areas declared using **TABLES** are **interface work areas** and should **only be** declared in the **global declaration part** of a program for the following purpose:

사용목적

The statement **TABLES** is required for **exchanging data between dynpro fields and the ABAP program**, if the fields were defined in a dynpro in the program by being taken from ABAP Dictionary .

In the dynpro event PBO, the content of the table work area is passed to **identically named dynpro fields**. In PAI, the system takes the data from identically named dynpro fields.

In executable programs, flat table work areas can be used to apply data that is provided for the event **GET table_wa** from an associated logical database.

TABLES is synonymous with the statement **NODES** for this purpose.

Thank you.

Contact information:

최창일 사내강사

