

# ABAP/4 Frequently Asked Question

## SAP R/3 Document : FAQ – ABAP/4 Dictionary

### ABAP/4 Dictionary

#### Q. What is a “DATA DICTIONARY”?

A data dictionary is a central source of information for the data in a data management system. Its main function is to support the creation and management of data definitions (or "metadata").

They enable a fast response to the following questions:

- What data is contained in the corporate databases?
- What are the attributes of this data (name, length, format, and so on)?
- What relationships exist between the different data objects?

#### Q. What functions does data dictionary perform?

In a data management system, the principal functions performed by a data dictionary are the following:

- management of data definitions
- provision of information for evaluations
- support for software development
- support for documentation
- ensuring that data definitions are flexible and up-to-date

#### Q. What is Relational Data Model?

A data model is a representation of a section of the real world employing formal structures. The relational data model employs only one such formal structure, referred to as a relation or table. All information on the section of the real world included in the model is shown in the relational model as values in tables.

#### Q. Explain ABAP/4 Dictionary?

The ABAP/4 Dictionary describes the logical structure of the objects used in application development and shows how they are mapped to structures in the underlying relational database. It makes central description of all data used in the system without redundancy possible.

The ABAP/4 Dictionary provides a logical view on the application data and the organization of this data in the underlying database system.

The ABAP/4 Dictionary is an integrated and active data dictionary. Integration in this case means that the ABAP/4 Dictionary is completely embedded in the ABAP/4 Development Workbench. Also, it is active in the run time environment.

#### Q. What information are directly taken from dictionary?

The ABAP/4 Repository Information System provides the user with a wide variety of options for analyzing the use of data and data dependencies. As a result, a high level of transparency of the data structure is obtained for the user. The displayed data also originates from the ABAP/4 Dictionary. A large part of the online help takes its information directly from the ABAP/4 Dictionary. In this way,

the ABAP/4 Dictionary helps to ensure that the online help is up-to-date and to reduce redundancy in the documentation.

The following information is taken directly from the ABAP/4 Dictionary:

- The information on fields displayed with F1 Help
- The possible entries for fields displayed with F4 Help
- The match code and help views search utilities.

### Q. What are the basic objects of ABAP/4 Dictionary?

The Basic elements are.

- Tables
- Data Elements
- Structures
- Domains

### Q. Explain each basic object of ABAP/4 Dictionary?

#### Tables :

In the ABAP/4 Dictionary, tables can be defined in a way, which is not database dependent. For this, fields of the table are defined with their database-independent (external) data types and lengths. In addition, table-specific settings, such as the buffering of the table or indexes for the table, can be defined.

When the table is activated, a physical table definition in the database is added to the table definition stored in the ABAP/4 Dictionary. The database-independent table definition from the ABAP/4 Dictionary is translated into a definition of the relevant database.

The attributes of table fields are distributed over three levels in the ABAP/4 Dictionary (fields, data elements, and domains), each attribute being assigned to only one of these levels, each field must refer to both a data element and a domain. This ensures the consistency of formally and semantically equivalent fields.

The minimum set of fields sufficient to uniquely identify each table line by their values is termed the **table key**.

The **technical settings** allow you to optimize storage space requirements and table access behavior for database tables. Via the technical settings you can determine how the table should be treated when it is created in the database, whether the table is buffered and whether changes to entries should be logged.

The most important parameters in the technical settings are:

**Data class:** The Data class establishes the physical area of the database (table space) in which your table is to be stored.

**Size category:** The Size category allows you to specify estimated space requirements for the table in the database. Efficient access to the data contained in a table can be supported by **Indexes**.

#### Index:

An index could be described as a copy of a database table reduced to specific fields. This copy always exists in sorted form. This sorting enables fast access to the data records of the table by means of binary search, for example. In order that the remaining fields can also be read, that is, those fields not contained

in the index, a pointer to the associated record of the actual table are included in the index.

A distinction is made between **primary and secondary indexes** to a table. The primary index consists of the key fields of the table and a pointer to the non-key-fields of the table. The primary index is generated automatically when a table is created and is created in the database at the same time as the table. It is also possible to define further indexes to a table in the ABAP/4 Dictionary, which are then referred to as secondary indexes. Where there are a number of different indexes to the same table, these are distinguished by means of a three-character **Index ID**.

If the values for the fields of an index uniquely identify a data record of the table, the index is referred to as a **Unique index**.

### **Sub structure:**

Instead of listing all the fields you wish the table to contain, you can choose to include the fields from another table or structure as a Substructure.

The Modification of SAP standard tables is rendered possible by append structures and customizing includes. These types of modifications are merged with the new standard SAP versions when Release upgrades are performed, ensuring that customer enhancements are not lost.

### **Data Element:**

A data element (semantic domain) describes the role played by a field in a technical context. Fields of the same semantic meaning refer to the same data element.

A data element carries field information valid for every field that refers to this data element, independently of the table in which a given field occurs. The semantic information includes the field's representation on the screen in the form of field texts, column captions in list outputs of the table contents and the format of the output via parameter IDs. This is also true for the online field documentation (that is, the text displayed in the field help (F1 help) for a screen field is taken from the corresponding data element).

### **Domain:**

A domain describes the technical characteristics of a table field. It specifies a value range, which describes allowed data values for the fields that refer to this domain.

The value range for a domain is defined by specifying a data type and a length. For example, specifying the data format NUMC and the field length 8 define the number of a person belonging to the university.

The ABAP/4 Dictionary provides two ways of further restricting the value range for a domain within a data type:

- By specifying **fixed values**. For example, the value range for the months in the year can be specified by listing all possible values (January, ..., December).
- By stipulating a **value table**. Only values contained in the relevant field of the value table may be entered in fields referring to this domain. A check is not made on the input mask for a field referring to this domain unless a corresponding foreign key has been defined for this field.

Both fixed values and a value table can be specified for a domain. As a result, fields referring to this domain can accept only those values that exist in both the value table and in the fixed values.

### **Structures:**

In the ABAP/4 Dictionary the construction of the data produced when calculations are carried out within programs or when data is transferred between programs can also be defined globally. This is achieved by defining a structure. A structure is defined in the ABAP/4 Dictionary like a table and can then be addressed from ABAP/4 programs.

While data in tables is stored permanently in the database, structures contain data only during the run time of a program.

Structures are defined in (almost) the same way as tables. The only difference is that no database table is generated. The same data elements and domains can be used in structures as are used in tables and it is even possible to include tables. As a result, it is possible to achieve a high level of consistency of data definitions even in complex programs, which link data taken from several places.

### **Q. Explain foreign keys in relational data model?**

The relational data model contains not only tables, but also relationships between tables. These relationships are defined in the ABAP/4 Dictionary by foreign keys. An important function of foreign keys is to support data integrity in the relational data model. Foreign key fields may assume only those values allowed by the check table, in other words, values occurring in the primary key of the check table.

These also form the basis for defining aggregate objects like views, matchcode etc.

### **Q. What are aggregate objects?**

Views, match codes and lock objects are called as aggregate objects since they are formed using several related tables.

**Views** a virtual table tailored to the needs of an application is needed, making it possible to directly access specific data.

**Match codes** is a tool to help you search for data records in the system. Match codes are an efficient and user-friendly search aid for cases where the key of a record is unknown.

**Lock objects:** Simultaneous accessing of the same data record by two users in the SAP system is synchronized by a lock mechanism. When dialog transactions are programmed, locks are set and released by calling certain function modules. These function modules are generated automatically from the definition of so-called lock objects in the ABAP/4 Dictionary.

### **Q. What are different types of views and explain them ?**

**Database views :** normally generated using joining tables. With this an equivalent view is created in the database.

**Projection views:** only for a single table and used to suppress certain fields of the table. Base tables are accessed via R/3 system.

**Help views:** used to display information on online help system. Base tables are accessed via the help processor in the R/3 system.

**Maintenance views:** are used to realize commercially relevant views on data. Base tables are accessed

via special maintenance modules.

Two further types exist for existing views. These view types cannot be used to create new views in Release 3.0.

**Structure views** are used to generate a structure from several logically connected tables. The structure is declared in the ABAP/4 program with the TABLES command and its fields can be accessed in the usual manner. Access is not permitted to these view structures with SQL. Structure views do not differ functionally from pure structures.

**Entity views** are used to represent entity types of the Data Modeler on tables of the ABAP/4 Dictionary. An entity view therefore shows how the attributes of an entity type are implemented in the ABAP/4 Dictionary with table fields. Entity views cannot be used in ABAP/4 programs.

### Q. Define Matchcode ?

**Matchcodes** are defined in two stages in the ABAP/4 Dictionary:

- The relevant tables and fields are stipulated in a Matchcode object. A matchcode object describes the set of all possible search paths for a search term.
- One or more Matchcode ID can be defined for a matchcode object. A Matchcode ID describes a special search path for a search term. The fields or combinations of fields via which the search is to take place are defined in the matchcode ID.

### Q. What are Type Groups?

You use type groups to store user-defined data types or constants in the ABAP/4 Dictionary for cross-program use.

In ABAP/4 program, you declare type groups with the TYPE-POOLS statement.

### Q. What are pooled and cluster tables?

A **table pool (or pool)** is used to combine several logical tables in the ABAP/4 Dictionary.

**Pooled tables** are logical tables that must be assigned to a table pool when they are defined.

Pooled tables can be used to store control data (such as screen sequences or program parameters).

Several pooled tables can be combined in a table pool. The data of these pooled tables are then stored in a common table in the database.

A **table cluster** combines several logical tables in the ABAP/4 Dictionary.

**Cluster tables** are logical tables that must be assigned to a table cluster when they are defined. Cluster tables can be used to store control data (such as screen sequences or program parameters). They can also be used to store temporary data or texts, such as documentation. Several cluster tables can be combined to form a table cluster. The data of the cluster tables is then stored in a single common table cluster in the database.