Abap data dictionary

Technology

ASSIGN BUSTER

The ABAP Dictionary centrally describes and manages all the data definitions used in the system. The ABAP Dictionary is completely integrated in the ABAP Development Workbench. All the other components of the Workbench can actively access the definitions stored in the ABAP Dictionary. The ABAP Dictionary supports the definition of user-defined types (data elements, structures and table types). You can also define the structure of database objects (tables, indexes and views) in the ABAP Dictionary. These objects can then be automatically created in the database with this definition.

The ABAP Dictionary also provides tools for editing screen fields, for example for assigning a field an input help (F4 help). Type definitions Structure Database objects Table DB table Data element Table type Tools Poss. values Screen F4 The most important object types in the ABAP Dictionary are tables, views, types (data elements, structures, table types), domains, search helps and lock objects. April 2001 9 BC - ABAP Dictionary ABAP Dictionary SAP AG ABAP Dictionary Purpose Data definitions (metadata) are created and managed in the ABAP Dictionary.

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. You can create the corresponding objects (tables or views) in the underlying relational database using these data definitions. The ABAP Dictionary therefore 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 a screen field an input help. What Information is Stored in the ABAP Dictionary? The most important object types in the ABAP Dictionary are tables, views, types, domains, search helps and lock objects. Tables [Page 13] are defined in the ABAP Dictionary independently of the database. A table having the same structure is then created from this table definition in the underlying database. Views [Page 97] are logical views on more than one table. The structure of the view is defined in the ABAP Dictionary.

A view on the database can then be created from this structure. Types [Page 136] are used in ABAP program. The structure of a type can be defined globally in ABAP programs. Changes to a type automatically take effect in all the programs using the type. Lock objects [Page 209] are used to synchronize access to the same data by more than one user. Function modules that can be used in application programs are generated from the definition of a lock object in the ABAP Dictionary. Different fields having the same technical type can be combined in domains [Page 161].

A domain defines the value range of all table fields and structure components that refer to this domain. The ABAP Dictionary also contains the information displayed with the F1 and F4 help for a field in an input template. The documentation about the field is created for a data element [Page 138] that describes the meaning of the contents of a table field. The list of possible input values that appears for the input help is created by a foreign key [Page 19] or a search help [Page 172]. Integration in the ABAP

Development Workbench The ABAP Dictionary is completely integrated in the ABAP Development Workbench.

The R/3 System works interpretatively, permitting the ABAP Dictionary to be actively integrated in the developmentenvironment. Instead of the original objects, the interpreters see only internal representations of these objects. These internal representations are adjusted automatically when the system finds that changes have been made in the ABAP Dictionary. This ensures that the screen and ABAP interpreters, input help, database interface, and development tools always access current data. 10 April 2001 SAP AG BC - ABAP Dictionary ABAP Dictionary

The following ABAP program lists the airline carriers (see Flight model [Page 302]) and carrier IDs contained in table SCARR. DATA: SCARR_TAB TYPE SCARR. SELECT * INTO SCARR_TAB FROM SCARR. WRITE: / SCARR_TAB-CARRID, SCARR_TAB-CARRNAME. ENDSELECT. Only structure SCARR_TAB is declared in the program. All the information about this structure, such as the field names, data types and field lengths, are copied from table SCARR, which is defined in the ABAP Dictionary. This information about table SCARR is called from the ABAP Dictionary when the program is generated.

This means that the source text of the program need not be adjusted when a change is made to table SCARR, for example when the length of a table field is changed. The next time the program is called, the system automatically determines that the structure of table SCARR has changed. The program is simply regenerated, thereby retrieving up-to-date information about table SCARR from the ABAP Dictionary. ? Development environment ?

Development environment ABAP Tools Data Modeler Screen Painter ABAP Dictionary ABAP Interpreter Dialog Control Interfaces Screen Interpreter

Runtime environment of the application Runtime environment of the application When you work on development projects, objects of the ABAP Dictionary can be changed any number of times before being activated [Page 237] and made available to the operative components of the system. Objects can have both an active and an inactive version in the ABAP Dictionary at the same time. Inactive ABAP Dictionary objects have no effect on the runtime system (ABAP processor, database interface). This permits greater changes to several objects without impairing the April 2001 11 BC - ABAP Dictionary ABAP Dictionary

SAP AG executability of the system. The objects can only be activated together when they have all been changed. 12 April 2001 SAP AG BC - ABAP Dictionary Tables Tables Tables can be defined independently of the database in the ABAP Dictionary. The fields of the table are defined with their (database-independent) data types and lengths. When the table is activated, a physical table definition is created in the database for the table definition stored in the ABAP Dictionary. The table definition is translated from the ABAP Dictionary to a definition of the particular database.

Database-independent Definition of the Tables in the ABAP Dictionary T1 T2 T3 ... Tn Activation program and DB UTILITY DB Definition of the tables in the database T1 T2 T3 Tn A table definition in the ABAP Dictionary contains the following components: · · · · Table fields [Page 14] define the field names and data types of the fields contained in the table Foreign keys [Page 19] define the relationships between the table and other tables. Technical settings

[Page 30] control how the table should be created in the database. Indexes [Page 61]: To speed up data selection, secondary indexes can be created for the table

The customer can modify SAP tables with append structures [Page 69] and customizing includes [Page 68]. This kind of modification ensures that the customer enhancements are automatically merged with the new versions of the SAP tables when there is a release upgrade. See also: Creating Tables [Page 72] Making Changes to Tables [Page 83] April 2001 13 BC - ABAP Dictionary Table Fields SAP AG Table Fields You must define the following for a table field in the ABAP Dictionary: · · · · · · Field name: The field name can have a maximum of 16 places and may contain letters, digits and underscores.

The field name must begin with a letter. Key flag: determines whether the field should belong to the table key. Field type: data type of the field in the ABAP Dictionary. Field length: number of valid places in the field. Decimal places: number of places after the decimal point, specifying numeric data types. Short text: short text describing the meaning of the field. You can also include [Page 16] the fields of a structure in the table. Assignment of the Data Type, Field Length and Short Text You can assign the data type [Page 242], length and short text in different ays: · · You directly assign the field a data type, field length (and if necessary decimal places) and short text in the table definition. You can assign the field a data element [Page 138]. The data type, field length (and decimal places) are determined from the domain of the data element. The short description of the data element is assigned to the field as a short text. Other Assignment Options · · · Check table: An input check for the field can be defined with a foreign key [Page 19]. This input check appears on all the screens in which the field is used. Search help assignment: A search help [Page 172] can be assigned to a field.

This search help defines the input help flow on all the screens in which the field is used. Reference field and reference table [Page 15]: You must specify the table field in which the corresponding unit of measure or currency can be found for fields containing quantities (data type QUAN) or currency amounts (data type CURR). See also: Creating Tables [Page 72] 14 April 2001 SAP AG BC - ABAP Dictionary Reference Fields and Reference Tables Reference Fields and Reference Tables You must specify a reference table for fields containing quantities (data type QUAN) or currency amounts (data type CURR).

This reference table must contain a field with the format for the currency key (data type CUKY) or unit of measure (data type UNIT). This field is called the reference field of the output field. The reference field can also reside in the table itself. A field is only assigned to the reference field at program runtime. For example, if a field is filled with currency amounts, the corresponding currency is determined from the assigned reference field, that is the value entered in this field at the moment defines the currency. Table Field 1 T1 Field 3 Field 2 (CURR)

Reference table Field 4 Field 5 (CUKY) T2 Field 7 Field 6 Reference field Runtime T1-Field 2 1, 500. 00 T2-Field 5 DEM Table SBOOK in the flight model [Page 302] contains all the flight bookings made by customers. Field FORCURAM contains the price of the booking in the customer’s currency. Field FORCURKEY of table SBOOK contains the corresponding currency key for this price. SBOOK is therefore the reference table for field FORCURAM and

FORCURKEY is the reference field for field FORCURAM. April 2001 15 BC - ABAP Dictionary Reference Fields and Reference Tables SAP AG

Includes In addition to listing the individual fields, you can also include the fields of another structure in tables [Page 13] and structures [Page 144]. Individual fields and includes can be mixed as required. Structure includes Table F1 F2 F3 F4 F5 F3 F4 F1 F2 F3 F4 F5 Database When an include is changed, all the tables and structures that include it are automatically adjusted. Structure A was included in table B. A new field is inserted in structure A. When structure A is activated, table B is adjusted to this change, that is the new field is also inserted there.

You can assign the include a group name [Page 148] with which the group of fields in the include can be addressed as a whole in ABAP programs. Includes can also be nested, that is structure A includes structure B which in turn includes another structure C, etc. The maximum nesting depth is limited to nine. The maximum length of a path of nested includes in a table or structure is therefore nine (the table/structure itself not included). 16 April 2001 SAP AG BC - ABAP Dictionary Reference Fields and Reference Tables Table/structure U1 Include U1 U2

Include U2 U3 Maximum depth = 9 Include U8 U9 Include U9 Only flat structures [Page 144] can be included. In a flat structure, every field either refers to a data element or is directly assigned a data type, length and possibly decimal places. Only structures may be included in a table. Tables, structures and views may be included in a structure. The length of the field names is more restricted in tables than in structures. In a table, a field name may not have more than 16 places, but in a structure up to 30 places are allowed for the field name.

A structure therefore can only be included in a table if none of the field names of the structure are longer than 16 places. The path of nested includes may only contain one table. Table TAB1 includes structure STRUCT1, which in turn includes structure STRUCT2. The path of the nested includes here only contains table TAB1. It is also possible to include TAB1 in a further structure STRUCT0, but no other table TAB2 may be included in TAB1 since in this case a path of nested includes would contain two tables (TAB1 and TAB2). See also: Inserting an Include [Page 85] April 2001 17 BC - ABAP Dictionary Named Includes SAP AG

Named Includes If an include [Page 16] is used to define a database table or structure, a name can be assigned to the included substructure. The group of fields in the include can be addressed as a whole in ABAP programs with this name. In ABAP programs, you can either access the fields directly with - or analogously with --. You can access the fields of the group as a whole with -. Structure PERSON includes structure ADDRESS with the name ADR. ADDRESS has a field CITY. With PERSON-ADR you can address all the fields in structure ADDRESS. The included field CITY can also be addressed with PERSON-CITY or PERSON-ADR-CITY.

You can include a structure more than once (e. g. in a period group). Since direct access by field name should be permitted here, the included field names must be renamed to ensure that they are unique. A suffix can be assigned to each group, extending the names of the group fields. The fields can then be addressed in ABAP programs with - or --. Structure PERSON includes structure ADDRESS twice. An address is the private address with suffix H and name ADRH. The other address is the business address with suffix W and name ADRW. You can access field CITY in the private address with PERSON-CITYH or PERSON-ADRH-CITY.

The functionality of the named includes in the ABAP Dictionary corresponds to the ABAP construction INCLUDE TYPE ... AS ... RENAMING ... . 18 April 2001 SAP AG BC - ABAP Dictionary Foreign Keys Foreign Keys You can define the relationships between tables in the ABAP Dictionary by creating foreign keys. Using foreign keys, you can easily create value checks for input fields. Foreign keys can also be used to link several tables in a view [Page 97] or a lock object [Page 209]. Field Assignment in the Foreign Key A foreign key links two tables T1 and T2 by assigning fields of table T1 to the primary key fields of table T2.

Foreign key fields Foreign key table T1 Field 1 Field 2 Field 3 Field 4 Primary key Check table Field 5 Field 6 T2 Field 7 Primary key Table T1 is called the foreign key table (dependent table) and table T2 the check table (referenced table). The pair of fields for the two tables must have the same data type and length. One field of the foreign key table therefore corresponds to each key field of the check table. This field is called the foreign key field. A foreign key permits you to assign data records in the foreign key table and check table.

One record of the foreign key table uniquely identifies one record of the check table using the entries in the foreign key fields. Check Field and Value Check One of the foreign key fields is marked as the check field. This means that the foreign key relationship is maintained for this field. April 2001 19 BC

- ABAP Dictionary Foreign Keys SAP AG When an entry is made in the check field, there is a check whether the check table contains a record with the key defined by the values in the foreign key fields. If this is so, the entry is valid. Otherwise the system rejects the entry.

Input template for foreign key table T1 Field1 Field2 Field3 Field4 1 3 Field5 1 1 2 3 3 3 4 4 Check table T2 Field6 1 3 1 1 2 3 1 2 Field7 Text 1 Text 2 Text 3 Text 4 Text 5 Text 6 Text 7 Text 8 Input is valid since there is a corresponding record in the check table In this example the entry Field2 = 2 and Field4 = 2 would be rejected since T2 does not contain a record with the key Field5 = 2 and Field6 = 2. If you do not want to check against all the key fields of the check table, you can exclude fields of the foreign key table from the assignment of the fields to the check table with generic and constant foreign keys [Page 22].

How the Input Check Works A SELECT statement is generated from the definition of the foreign key. If an entry is made in the check field, this SELECT statement is submitted. If a suitable record of the check table is found, the entry is valid. Otherwise the entry is rejected. The corresponding SELECT statement has the following form for the foreign key table shown in the above graphic: SELECT * FROM T2 WHERE T2-FIELD5 = T1-FIELD2 AND T2-FIELD6 = T1-FIELD4. A screen entry for check field Field2 is therefore only valid if the check table contains a record with the entries made in the screen for Field2 and Field4 as key.

Table SBOOK in the flight model [Page 302] contains the customer’s flight bookings for a carrier. The flight bookings can be made by a travel agency or directly at the carrier’s sales counter. If the booking was made at a counter, its number is stored together with the booking in field COUNTER in table SBOOK. 20 April 2001 SAP AG BC - ABAP Dictionary Foreign Keys You must make sure that only correct counter numbers can be entered. All the counters are entered in table SCOUNTER. The necessary value check can be defined by creating a foreign key for check field COUNTNUM. Foreign key fields Foreign key table SBOOK

MANDT CARRID CONNID FLDATE CUSTOMID ... COUNTER ... CANCELED Check field Check table SCOUNTER MANDT CARRID COUNTNUM AIRPORT Key fields See also: Multi-Structured Foreign Keys [Page 29] Semantic Attributes of Foreign Keys [Page 24] Creating Foreign Keys [Page 75] April 2001 21 BC - ABAP Dictionary Generic and Constant Foreign Keys SAP AG Generic and Constant Foreign Keys It is not always advisable to check a foreign key against all the key fields of the check table. This is true for example for time-dependent check tables and for check tables whose version number is a component of the key.

You can use generic foreign keys in these cases. Fields are excluded from the assignment to the key fields of the check table here. The check is only against the remaining key fields. You can also assign a constant value to a key field of the check table. In this case you only have to check against the specified constant. You can use this check if only records of the check table which contain a constant value in this key field are valid. Foreign key table FTAB Field 6 Field 7 Field 8 Field 9 Generic * Constant K Check table PTAB Field 1 Field 2 Field 3 Field 4 Field 5

Primary key The corresponding SELECT statement for the screen check has the following form for the foreign key definition in the graphic: SELECT *

FROM PTAB WHERE PTAB-FIELD1 = FTAB-FIELD6 AND PTAB-FIELD3 = FTABFIELD8 AND PTAB-FIELD4 = ‘ K’. An entry is only valid in check field Field6 if a record of check table PTAB exists containing the input value for Field6 in PTAB-Field1, the input value for Field8 in PTAB-Field3 and constant K in PTAB-Field4. 22 April 2001 SAP AG BC - ABAP Dictionary Generic and Constant Foreign Keys

Input template for foreign key table FTAB Field 6 Field 7 Field 8 Field 9 3 30 1 B Check table PTAB Field 1 Field 2 Field 3 Field 4 Field 5 1 1 2 3 3 3 4 4 1 1 1 2 1 2 1 2 1 3 1 1 2 3 3 4 A B A K A A C C Text 1 Text 2 Text 3 Text 4 Text 5 Text 6 Text 7 Text 8 Input is valid since Field 7 and Field 9 were removed from the assignment The values entered on the screen for Field7 and Field9 are meaningless when checking against the check table. An entry with Field6 = 1, Field8 = 3 and Field9 = B would not be valid in this case since there is no record with PTAB-Field1 = 1, PTAB-Field3 = 3 and PTAB-Field4 = K in the check table!

April 2001 23 BC - ABAP Dictionary Semantic Attributes of Foreign Keys SAP AG Semantic Attributes of Foreign Keys A foreign key describes a relationship between two tables. You can define this relationship more precisely by specifying the cardinality [Page 25] and type of foreign key fields [Page 26]. This information is optional and is primarily for documentary purposes. In particular, the definitions of the cardinality and type of the foreign key fields are not used in the value check for the foreign key. The definition of the semantic attributes is only sed in the following cases: · If Key fields of a text table is selected as the type of the foreign key fields, the foreign key table is considered to be the text table [Page 27] for the check table. If a screen field is checked against a table, the key entries of the check table are normally displayed in the input help (F4 help) for this field. If there is a text table for the check table, each key entry displayed is enhanced with an explanatory text (contents of the first character-like field of the text table) in the user’s logon language.

Tables can only be included in a help view [Page 115] or maintenance view [Page 117] if they are linked with a foreign key. It only makes sense to create such a help or maintenance view if for each record in the primary table of the view there is no more than one corresponding record in each secondary table of the view. The system therefore checks if the foreign key with which the tables were linked in the view have suitable cardinalities when it creates a maintenance or help view. See also Restrictions for Maintenance and Help Views [Page 119]. The foreign key between tables SBOOK and SCOUNTER ensures that only existing counters can be entered in field COUNTER (counter at which the flight was booked). See the example in Foreign Keys [Page 19] . A booking can be made at either a travel agency or at the carrier’s sales counter. If the booking is made at a travel agency, the field COUNTER of table SBOOK remains empty. The foreign key fields do not have to be filled, that is the left side of the cardinality is C. Any number of bookings may be made at each counter.

There may therefore be any number of entries (bookings) in foreign key table SBOOK for each record of the check table SCOUNTER. The right side of the cardinality is therefore CN. Of course several bookings can be made for the same carrier at a counter. These bookings do not differ in their foreign key fields (MANDT, CARRID, COUNTER). The entries in the foreign key fields therefore do not uniquely identify an entry in the foreign key table SBOOK (a booking). The foreign key fields therefore have the type No key fields/candidates. 24 April 2001 SAP AG BC - ABAP Dictionary Cardinality Cardinality

The cardinality (n: m) describes the foreign key relationship with regard to the number of possible dependent records (records of the foreign key table) or referenced records (records of the check table). The left side (n) of the cardinality is defined as follows: · · n= 1: There is exactly one record assigned to the check table for each record of the foreign key table. n= C: The foreign key table may contain records which do not correspond to any record of the check table because the foreign key field is empty. This can occur for example if the field of the foreign key table is optional, in which case it does not have to be filled. = 1: There is exactly one dependent record for each record of the check table. m= C: There is at most one dependent record for each record of the check table. m= N: There is at least one dependent record for each record of the check table. m= CN: There may be any number of dependent records for each record of the check table. The right side (m) of the cardinality is defined as follows: · · · · April 2001 25 BC - ABAP Dictionary Type of Foreign Key Fields SAP AG Type of Foreign Key Fields The Type of foreign key fields describes what the foreign key fields in the foreign key table mean.

The following types of foreign key field can be defined: · No key fields/candidates: The foreign key fields are neither primary key fields of the foreign key table nor do they uniquely identify a record of the foreign key table (key candidates). For this reason, the foreign key fields do not (partially) identify the foreign key table. Key fields/candidates: The foreign key fields are either primary key fields of the foreign key table or they already uniquely identify a record of the foreign key table (key candidates). The foreign key fields therefore (partially) identify the foreign key table.

Key fields of a text table: The foreign key table is a text table [Page 27] for the check table, that is the key of the foreign key table only differs from the key of the check table in that it has an additional language key field. This is a special case of the type Key fields/candidates. · · 26 April 2001 SAP AG BC - ABAP Dictionary Text Tables Text Tables Table A is a text table of table B if the key of A comprises the key of B and an additional language key field (field of data type LANG). Table A may therefore contain explanatory text in several languages for each key entry of B.

To link the key entries with the text, text table A must be linked with table B using a foreign key. Key fields of a text table must be selected here for the type of foreign key fields (see Semantic Attributes of Foreign Keys [Page 24]). Table B Key fields K1 and K2 K1 ... 1 1 ... K2 ... 1 2 ... F1 ... XX YY ... F2 ... YY XX ... Text table A for B Key fields K1, K2 and L (type LANG) K1 ... 1 1 1 1 ... K2 ... 1 1 2 2 ... L ... DE EN DE EN ... TEXT ... Text 1 (German) Text 1 (English) Text 2 (German) Text 2 (English) ... Text foreign key

If table B is the check table of a field, the existing key entries of table B are displayed as possible input values when the input help (F4) is pressed. The explanatory text (contents of the first character-like non-key-field of text table A) is also displayed in the user's logon language for each key value in table B. April 2001 27 BC - ABAP Dictionary Text Tables SAP AG Hit list if user logs on in English K1 ... 1 1 K2 ... 1 2 ... Text ... Text1 (English) (English)

Text2 (English) (English) ... Maintenance screen Field 1 Field 2 ... Call the input help Field is checked against table B

Only one text table can be created for table B! The system checks this when you attempt to activate a table with text foreign keys for B. 28 April 2001 SAP AG BC - ABAP Dictionary Multi-Structured Foreign Keys Multi-Structured Foreign Keys When you define a foreign key, a field of the work area that is not contained in the foreign key table can also be assigned to a check table (for example a field of another table). This is possible for all fields except for the check field. Table T2 is the check table of foreign key table T1. Field F of the work area is assigned to key field Field6 of check table T2.

Foreign key table T1 Field 1 Field 2 Field 3 Field 4 Primary key Field F of work area Check table T2 Field 5 Field 6 Field 7 Primary key The corresponding SELECT statement for the input check is then: SELECT * FROM T2 WHERE T2-FIELD5 = T1-FIELD2 AND T2-FIELD6 = F. If an entry is made in field T1-Field2 (check field), this SELECT statement will be submitted. If a corresponding record is found, the entry is valid; otherwise it is rejected. If a field that is not contained in the foreign key table is assigned to a field of the check table, this field must be filled at the time of the input check.

Otherwise the check always fails, and no values can be entered in the check field. April 2001 29 BC - ABAP Dictionary Technical Settings SAP AG Technical Settings The technical settings of a table define how the table will be handled when it is created in the database, that is whether the table will be buffered and whether changes to data records of the table will be logged. The most important parameters are: · · Data class: The data class [Page 31] defines the physical area of the database (tablespace) in which the table should be created. Size category: The size category [Page 32] defines the size of the extents created for the table.

When the table is created in the database, the required information about the memory area to be selected and the extent size is determined from the technical settings. · · Buffering permission: The buffering permission [Page 33] defines whether the table may be buffered. Buffering type: If the table may be buffered, you must define a buffering type (full, singlerecord, generic). The buffering type [Page 34] defines how many table records are loaded into the buffer when a table entry is accessed. Logging: This parameter defines whether changes to the table entries should be logged.

If logging [Page 41] is switched on, each change to a table record is recorded in a log table. · The Convert to transparent table flag (transparent flag [Page 42]) is also displayed for pooled tables or for tables which were converted into transparent tables earlier on with this flag. See also: Maintaining Technical Settings [Page 77] Buffering Database Tables [Page 43] 30 April 2001 SAP AG BC - ABAP Dictionary Data Class Data Class If you choose the data class correctly, your table is automatically assigned to the correct area (tablespace or DBspace) of the database when it is created.

Each data class corresponds to a physical area in which all the tables assigned to this data class are stored. There are the following data classes: · · · APPL0 (master data): Data which is seldomly changed. An example of master data is the data contained in an address file, such as the name, address and telephone number. APPL1 (transaction data): Data that is frequently changed. An example of transaction data is the goods in a warehouse, which change after each purchase order. APPL2 (organizational data): Customizing data that is defined when the system is installed and seldomly changed.

An example is the table with country codes. Two further data classes, USR and USR1, are provided for the customer. These are for user developments. The tables assigned to these data classes are stored in a tablespace for user developments. Tables in the ABAP Dictionary Master data Table 1 Table 3 Organizational data Table 2 Transaction data Table 4 Table 7 System data Table 5 Table 6 Tablespace master data Table 1 Table 3 Tablespace Org. data Table 2 Tablespace Trans. data Table 4 Table 7 Tablespace System data Table 5 Table 6 Database April 2001 31 BC - ABAP Dictionary Size Category SAP AG Size Category

The size category defines the expected space required for the table in the database. You can choose a size category from 0 to 4 for your table. Each category is assigned a certain fixed memory size in the database, which depends on the database system used. When a table is created, initial space (an Initial Extent) is reserved in the database. If more space is required at a later time due to data entries, additional memory will be added depending on the selected size category. Technical settings Size category TABA 1 3 4 TABB TABC Initial First Second Extent Extent Extent TABA TABB TABC ... ...

Database Selecting the correct size category prevents a large number of very small extents from being created for a table. It also prevents space from being wasted if extents which are too large are created. 32 April 2001 SAP AG BC - ABAP Dictionary Buffering Permission Buffering Permission You must define whether and how a table is buffered in the technical settings for the table. There are three possibilities here: · Buffering not permitted: Table buffering is not permitted, for example because application programs always need the most recent data from the table or the table is changed too frequently.

Buffering permitted but not activated: Buffering is permitted from the business and technical points of view. Applications which access the table execute correctly with and without table buffering. Whether or not table buffering will result in a gain in performance depends on the table size and access profile of the table (frequency of the different types of table access). Table buffering is deactivated because it is not possible to know what these values will be in the customer system. If table buffering would be advantageous for the table size and access profile of the table, you can activate it in the customer system at any time.

Buffering activated: The table should be buffered. In this case you must specify a buffering type [Page 34]. · · See also: Buffering Database Tables [Page 43] Which Tables Should be Buffered? [Page 53] April 2001 33 BC - ABAP Dictionary Buffering Types SAP AG Buffering Types The buffering type defines which table records are loaded into the buffer of the application server when a table record is accessed. There are the following buffering types: · · Full buffering [Page 35]: All the records of the table are loaded into the buffer when one record of the table is accessed.

Generic buffering [Page 37]: When a record of the table is accessed, all the records having this record in the generic key fields (part of the table key that is left-justified, identified by specifying a number of key fields) are loaded into the buffer. Single-record buffering [Page 39]: Only the records of a table that are really accessed are loaded into the buffer. · See also: Buffering

Database Tables [Page 43] 34 April 2001 SAP AG BC - ABAP Dictionary Full Buffering Full Buffering With full buffering, either the entire table is in the buffer or the table is not in the buffer at all.

All the records of the table are loaded into the buffer when one record of the table is read. In this example, a program reads the record highlighted in red from table SCOUNTER. If the table is fully buffered, all the records of the table are loaded into the buffer. Database table SCOUNTER MANDT CARRID COUNTNUM AIRPORT Buffer contents 001 001 001 001 001 001 001 001 001 001 001 001 001 001 AA BA BA BA BA LH LH LH LH LH LH LH LH UA 00000001 00000001 00000002 00000003 00000004 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000001 ACA ACE BER LCY LHR BER DEN FRA LCY LGW LHR MUC RTM HAM 01 001 001 001 001 001 001 001 001 001 001 001 001 001 AA BA BA BA BA LH LH LH LH LH LH LH LH UA 00000001 00000001 00000002 00000003 00000004 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000001 ACA ACE BER LCY LHR BER DEN FRA LCY LGW LHR MUC RTM HAM Application server SELECT * FROM SCOUNTER WHERE MANDT = ‘ 001’ AND CARRID = ‘ LH’ AND COUNTNUM = '00000004'. The buffered data records are sorted in the buffer by table key. Accesses to the buffered data can therefore only analyze field contents up to the last specified key field for restricting the dataset to be searched.

The left-justified part of the key should therefore be as large as possible in such accesses. For example, if you do not define the first key field, the system has to scan the full table. In this case direct access to the database can be more efficient if the database has suitable secondary indexes [Page

61]. When Should you Use Full Buffering? When deciding whether a table should be fully buffered, you should take into account the size of the table, the number of read accesses, and the number of write accesses. Tables best suited to full buffering are small, read frequently, and rarely written.

Full buffering is recommended in the following cases: April 2001 35 BC - ABAP Dictionary Full Buffering · SAP AG Tables up to 30 KB in size. If a table is accessed frequently, but all accesses are read accesses, this value can be exceeded. However, you should always pay attention to the buffer utilization. Larger tables where large numbers of records are frequently accessed. If these mass accesses can be formulated with a very selective WHERE condition using a database index [Page 61], it could be better to dispense with buffering.

Tables for which accesses to non-existent records are frequently submitted. Since all the table records reside in the buffer, the system can determine directly in the buffer whether or not a record exists. · · 36 April 2001 SAP AG BC - ABAP Dictionary Generic Buffering Generic Buffering With generic buffering, all the records in the buffer whose generic key fields match this record are loaded when one record of the table is accessed. The generic key is a part of the primary key of the table that is left-justified. In this example, the record highlighted in red is read by a program from table SCOUNTER.

If the table is generically buffered, all the records read whose generic key fields (MANDT and CARRID) agree are loaded into the buffer. Database table SCOUNTER MANDT CARRID COUNTNUM AIRPORT Buffer contents 001 001 001 001 001 001 001 001 LH LH LH LH LH LH LH LH 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 BER DEN

FRA LCY LGW LHR MUC RTM 001 001 001 001 001 001 001 001 001 001 001 001 001 001 AA BA BA BA BA LH LH LH LH LH LH LH LH UA 00000001 00000001 00000002 00000003 00000004 00000001 00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000001

ACA ACE BER LCY LHR BER DEN FRA LCY LGW LHR MUC RTM HAM Application server Generic key SELECT * FROM SCOUNTER WHERE MANDT = ‘ 001’ AND CARRID = ‘ LH’ AND COUNTNUM = '00000004'. When Should you Use Full Buffering? A table should be buffered generically if only certain generic areas of the table are normally needed for processing. Client-specific, fully-buffered tables are automatically generically buffered since normally it is not possible to work in all clients at the same time on an application server. The client field is the generic key. Language-specific tables are another example where generic buffering is recommended.

In general, only records of one language will be needed on an application server. In this case, the generic key includes all the key fields up to and including the language field. How Should you Define the Generic Key? In generic buffering, it is crucial to define a suitable generic key. April 2001 37 BC - ABAP Dictionary Generic Buffering SAP AG If the generic key is too small, the buffer will contain a few very large areas. During access, too much data might be loaded in the buffer. If the generic key is too large, the buffer might contain too many small generic areas.

These can reduce buffer performance since there is an administrative entry for every buffered generic area. It is also possible that too many accesses will bypass the buffer and go directly to the database, since they do not fully define the generic key of the table. If there are only a few records in each generic area, it is usually better to fully buffer the table. Only 64 bytes of the generic key are used. You can specify a longer generic key, but the part of the key exceeding 64 bytes is not used to create the generic areas. Access to Buffered Data

It only makes sense to generically buffer a table if the table is accessed with fully-specified generic key fields. If a field of the generic key is not assigned a value in a SELECT statement, it is read directly from the database, bypassing the buffer. If you access a generic area that is not in the buffer with a fully-specified generic key, you will access the database to load the area. If the table does not contain any records in the specified area (" No record found"), this area in the buffer is marked as non-existent. It is not necessary to access the database if this area is needed again. 8 April 2001 SAP AG BC - ABAP Dictionary Single-Record Buffering Single-Record Buffering With single-record buffering, only the records that are actually read are loaded into the buffer. Single-record buffering therefore requires less storage space in the buffer than generic and full buffering. The administrative costs in the buffer, however, are greater than for generic or full buffering. Considerably more database accesses are necessary to load the records than for the other buffering types. In this example, the record highlighted in red is read by a program from table SCOUNTER.

If single-record buffering is selected for the table, only the record that was read is loaded into the buffer. Database table SCOUNTER MANDT CARRID COUNTNUM AIRPORT Buffer contents 001 LH 00000004 LCY 001 001 001 001 001 001 001 001 001 001 001 001 001 001 AA BA BA BA BA LH LH LH LH LH LH LH LH UA 00000001 00000001 00000002 00000003 00000004 00000001

00000002 00000003 00000004 00000005 00000006 00000007 00000008 00000001 ACA ACE BER LCY LHR BER DEN FRA LCY LGW LHR MUC RTM HAM Application server SELECT SINGLE FROM SCOUNTER WHERE MANDT = ‘ 001’ AND CARRID = ‘ LH’ AND COUNTNUM = '00000004'.

When Should you Use Single-Record Buffering? Single-record buffering should be used particularly for large tables where only a few records are accessed with SELECT SINGLE. The size of the records being accessed should be between 100 and 200 KB. Full buffering is usually more suitable for smaller tables that are accessed frequently. This is because only one database access is necessary to load such a table with full buffering, whereas several database accesses are necessary for single-record buffering. Access to Buffered Data All accesses that are not submitted with SELECT SINGLE go directly to the database, bypassing the buffer.

This applies even if the complete key is specified in the SELECT statement. April 2001 39 BC - ABAP Dictionary Single-Record Buffering SAP AG If you access a record which is not yet buffered with SELECT SINGLE, there is a database access to load the record. This record is marked in the buffer as non-existent if the table does not contain a record with the specified key. This prevents another database access when accessing the table at a later time with the same key. 40 April 2001 SAP AG BC - ABAP Dictionary Logging Logging Using the logging flag you can define whether changes to the data records of a table should be logged.

If logging is switched on, each change to an existing data record (with UPDATE, DELETE) by the user or application program is recorded in the database in a log table (DBTABPRT). ABAP Dictionary Log TAB Application transaction TAB Change a record Field 1 Field 2 Field 3 System profile ... rec/client = ALL ... TAB Field 1 Field 2 Field 3 Log table Database To switch on logging, the R/3 System must be started with a profile containing parameter rec/client. This parameter defines whether all clients or only selected clients should be logged. The parameter can have the following values: rec/client = ALL Log all clients. ec/client = 000[,...] Log the specified clients. rec/client = OFF Do not log. Logging slows down accesses that change the table. First of all, a record must be written in the log table for each change. Secondly, a number of users access this log table in parallel. This can cause lock situations although the users are working with different application tables. Logging is independent of the update. The existing logs can be displayed with Transaction Table History (SCU3). April 2001 41 BC - ABAP Dictionary Converting Pooled Tables to Transparent Tables SAP AG Converting Pooled Tables to Transparent Tables