INTERNATIONAL STANDARD

ISO/IEC 7816-7

First edition
1999-03-01

Identification cards — Integrated circuit(s) cards with contacts —

Part 7:
Interindustry commands for Structured Card Query Language (SCQL)

Cartes d'identification — Cartes à circuit(s) intégré(s) à contacts —

Partie 7: Commandes intersectorielles pour langage d'interrogation de carte structurée (SCQL)

Reference number
ISO/IEC 7816-7:1999(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 7816-7 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommitte SC 17, *Identification cards and related devices.*

ISO/IEC 7816 consists of the following parts, under the general title *Identification cards — Integrated circuit(s) cards with contacts*:

— *Part 1: Physical characteristics*

— *Part 2: Dimensions and location of the contacts*

— *Part 3: Electronic signals and transmission protocols*

— *Part 4: Interindustry commands for interchange*

— *Part 5: Numbering system and registration procedure for application identifiers*

— *Part 6: Interindustry data elements*

— *Part 7: Interindustry commands for Structured Card Query Language (SCQL)*

— *Part 8: Security related interindustry commands*

Annex A of this part of ISO/IEC 7816 is for information only.

Introduction

This part of ISO/IEC 7816 is one of a series of standards describing the parameters for integrated circuit(s) cards with contacts and the use of such cards for international interchange.

These cards are identification cards intended for information exchange negotiated between the outside and the integrated circuit in the card. As a result of an information exchange, the card delivers information (computation results, stored data), and/or modifies its content (data storage, event memorization).

During the preparation of this part of ISO/IEC 7816, information was gathered concerning relevant patents upon which application of this part of ISO/IEC 7816 might depend. Relevant patents were identified in France, the patent holder is Gemplus. However, ISO cannot give authoritative or comprehensive information about evidence, validity or scope of patents or like rights.

The patent holder has stated that licenses will be granted in appropriate terms to enable application of this part of ISO/IEC 7816, provided that those who seek licenses agree to reciprocate.

Further information is available from

GEMPLUS
B.P. 100
13881 GEMENOS CEDEX
FRANCE

Identification cards — Integrated circuit(s) cards with contacts —

Part 7:
Interindustry commands for Structured Card Query Language (SCQL)

1 Scope

This part of ISO/IEC 7816 specifies

— the concept of a SCQL database (SCQL = Structured Card Query Language based on SQL, see ISO 9075) and

— the related interindustry enhanced commands.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 7816. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 7816 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9075:1992, *Information technology — Database languages — SQL2.*

ISO/IEC 7816-4:1995, *Information technology — Identification cards — Integrated circuit(s) cards with contacts — Part 4: Interindustry commands for interchange.*

ISO/IEC 7816-6:1996, *Identification cards — Integrated circuit(s) cards with contacts — Part 6: Interindustry data elements.*

3 Terms and definitions

For the purposes of this part of ISO/IEC 7816, the following definitions apply.

3.1
database basic user
SCQL user with no inherent rights

3.2
database file
structured set of database objects (tables, views, dictionaries) representing the content of a database

3.3
database object owner
SCQL user with the special right to create and drop objects and to manage privileges on these objects

3.4
database owner
initial SCQL user which manages objects and users of the database

3.5
dictionary
view on a system table

3.6
system table
table maintained by the card for managing the database structure and database access

3.7
table
database object with a unique name and structured in columns and rows

3.8
view
logical subset of a table

4 Symbols (and abbreviated terms)

For the purposes of this part of ISO/IEC 7816, the following abbreviations apply:

APDU	Application protocol data unit
API	Application programming interface
DB	Database
DB_O	Database owner
DBBU	Database basic user
DBF	Database file
DBOO	Database object owner
DF	Dedicated file
DO	Data object
ICC	Integrated circuit(s) card
IFD	Interface device
MF	Master file
SCQL	Structured card query language
SQL	Structured query language
TLV	Tag, length, value

5 SCQL database concept

5.1 SCQL database

A database in a card according to this part of ISO/IEC 7816 is called a SCQL database (SCQL = Structured Card Query Language), since the commands for accessing are based on SQL-functionality (see ISO 9075) and coded according to the principles of interindustry commands as defined in ISO/IEC 7816-4. The database itself is a structured set of database objects called a database file DBF. Under a DF there shall be not more than one DBF which is accessible after selection of the respective DF. A database may be also directly attached to the MF.

Fig.1 shows an example for the embedding of a database in the card.

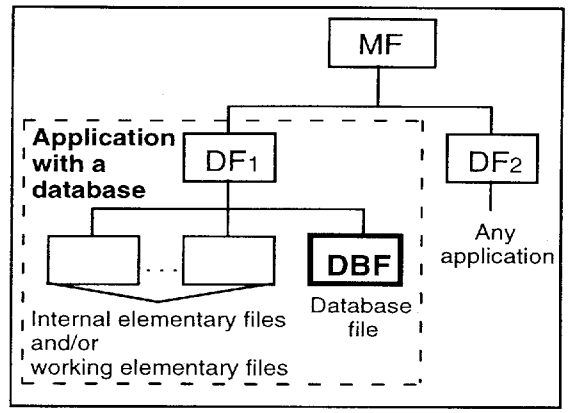

Figure 1 — Application with a database in a multi-application card (example)

An application system may interwork with a SQL database as well as with a SCQL database using the same SQL-API (API = Application Programming Interface). Thus, a card carrying a SCQL database may appear as a part of a distributed SQL database environment. Fig. 2 shows a typical SQL configuration with a card integrated in the system design.

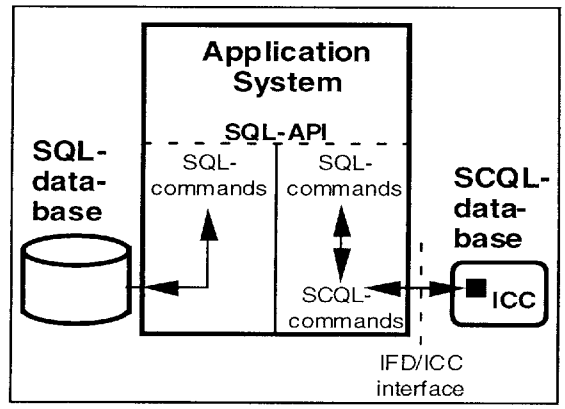

Figure 2 — SCQL database as part of a distributed SQL database environment (example)

5.2 SCQL tables

A SCQL database contains objects called tables, views and dictionaries. Each object can be referenced by a unique identifier.

A table is a structured data object with a unique name within a database. It consists of named columns and a sequence of rows. The number of rows may be conceptually unlimited (i.e., only restricted by the available memory space in the card), or limited. The table and the main characteristics are shown in fig. 3.

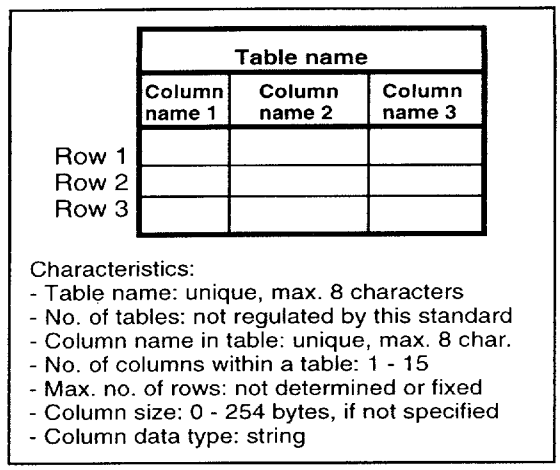

Figure 3 — SCQL table (example) and its main characteristics

After creation the table structure is persistent, i.e. neither an existing column can be withdrawn nor a new column can be inserted. On a table the following actions can be performed:

— read (select)

— insert

— update

— delete.

5.3 SCQL views

A view is a logical subset of a table, which defines the part of the table accessible. Two types of views are to be distinguished:

— a view (see fig. 4), which by definition fixes the accessible columns, is called in this context a static view and

— a view (see fig. 5), which restricts the access to those rows whose contents matches defined conditions (e.g. to rows the value of which is greater '20'), is called in this context a dynamic view.

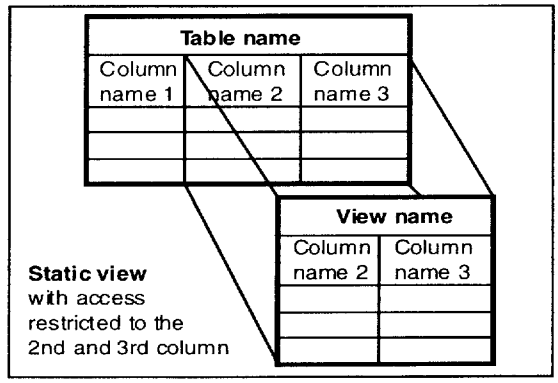

Figure 4 — SCQL static view (example)

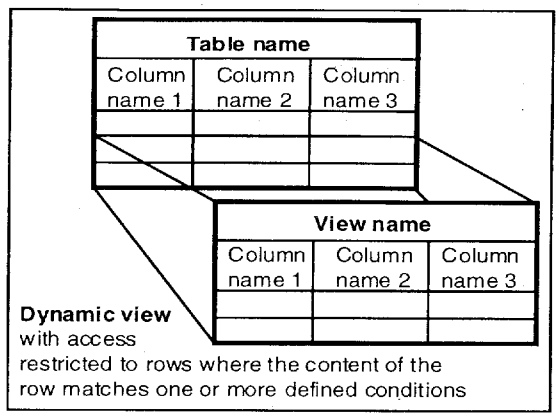

Figure 5 — SCQL dynamic view (example)

A combination of static view and dynamic view in the same view definition is also possible.

A view has like a table a unique name in a SCQL database. Several views may be defined on the same table.

On a view the following actions can be performed:

— read (select)

— update.

5.4 SCQL system tables and dictionaries

A system table is maintained by the card and contains information necessary to manage the database structure and access. There are three system tables:

— the object description table (name *O)

— the user description table (name *U)

— the privilege description table (name *P)

The object description table contains information about the tables and views stored in the database.

The user description table contains information about the users which have access to the database.

The privilege description table contains information about the privileges onto the database tables and views. Privileges describe which tables and views can be accessed by which users, and which actions can be performed by those users on the respective table or view.

The figures 6 - 8 show the system tables with their mandatory columns.

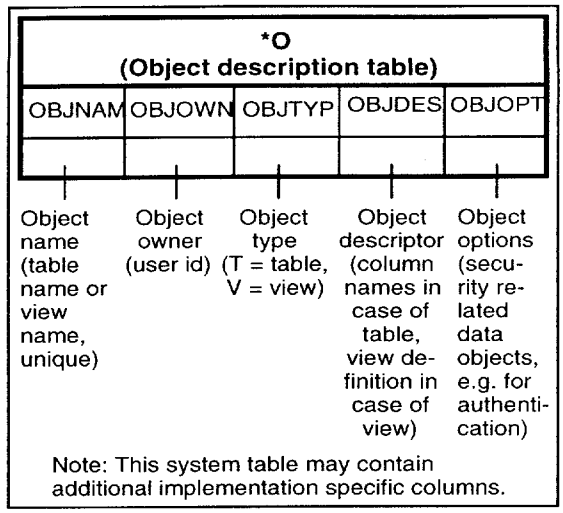

Figure 6 — Object description table

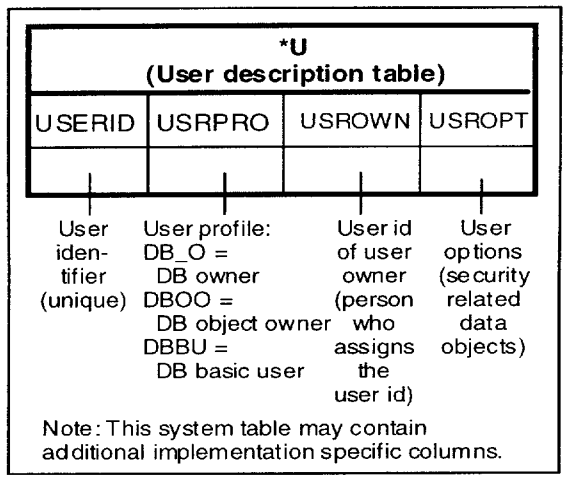

Figure 7 — User description table

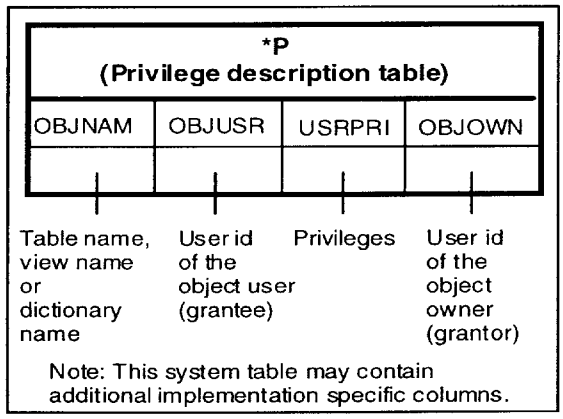

Figure 8 — Privilege description table

For access to the information contained in the system tables, views on these system tables can be created. A view on a system table is called a SCQL dictionary. The only action which a user can perform on a dictionary is reading (select).

5.5 SCQL user profiles

SCQL user profiles are characterized by special permissions. A user profile is attached to a user identifier stored in the user description table. Table 1 shows the profiles and the attached permissions.

Table 1 — SCQL user profiles and attached permissions

Profile	User	Permission
DB_O	Database owner	- Adding/dropping of users with profile DBOO or DBBU - Creation/deletion of objects (tables/views) - Granting/revoking of privleges for objects owned - Creation/deletion of dictionaries with access to all rows in the system tables - Access to objects not owned according to the privileges granted
DBOO	Database object owner	- Adding/dropping of users with profile DBBU - Creation/deletion of objects (tables/views) - Granting/revoking of privileges for objects owned - Creation/deletion of dictionaries with access to rows where the DBOO is registered as OBJOWN in *O, USROWN in *U or OBJOWN in *P - Access to objects not owned according to the privileges granted
DBBU	Database basic user with specific user id or the general user id PUBLIC	- Access to objects according to the privileges granted

NOTE A user with the profile DB_O can only be inserted in the user description table during the SCQL database installation.

6 SCQL related commands

6.1 General aspects

The 'Structured Card Query Language (SCQL)' is based on the functionality of the standardized 'Structured Query Language (SQL)'. SQL statements are mapped onto SCQL operations within the PERFORM SCQL OPERATION command (see fig. 9 and table 2).

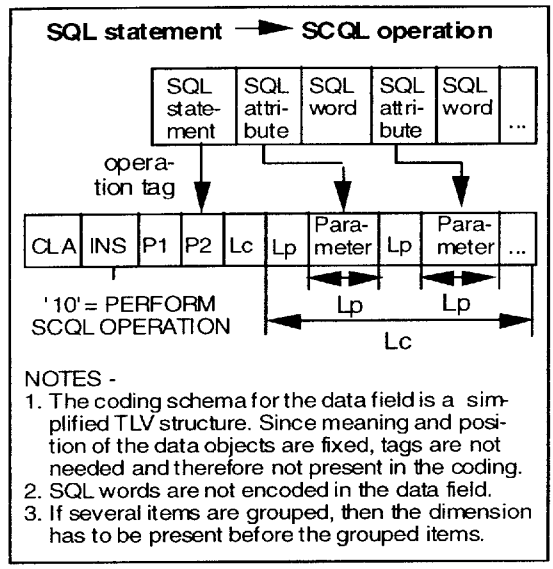

Figure 9 — Mapping principle of a SQL statement onto a SCQL operation

The mandatory parameters of a command occur always in the sequence defined in the related command table. Their tag is therefore not present. The optional parameters are - if not indicated otherwise - presented in TLV format.

As well as the PERFORM SCQL OPERATION command two other commands belong to the SCQL environment, but may be used also outside an SCQL environment:

— the PERFORM TRANSACTION OPERATION command (see fig. 10) and

— the PERFORM USER OPERATION command (see fig. 10).

6.2 Grouping and encoding of commands

The SCQL related commands can be grouped as shown in fig. 10.

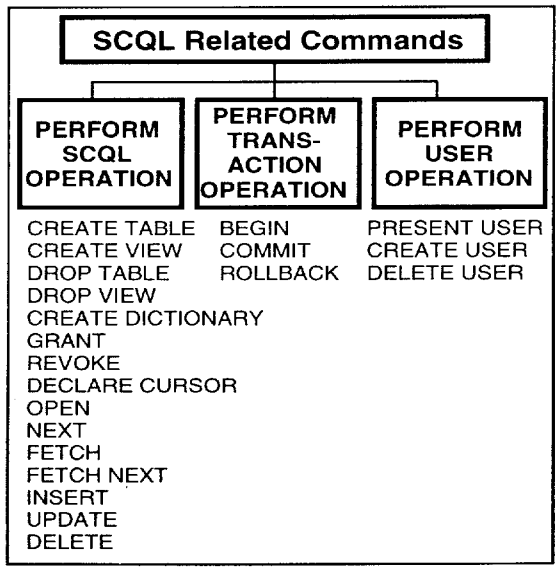

Figure 10 — SCQL related commands

For the commands defined in this part of the standard the instruction codes and the coding of the respective operations are shown in table 2.

Table 2 — Instruction codes and operations

INS-code	Meaning
'10'	**PERFORM SCQL OPERATION** P2 coding and meaning: '80' = CREATE TABLE '81' = CREATE VIEW '82' = CREATE DICTIONARY '83' = DROP TABLE '84' = DROP VIEW '85' = GRANT '86' = REVOKE '87' = DECLARE CURSOR '88' = OPEN '89' = NEXT '8A' = FETCH '8B' = FETCH NEXT '8C' = INSERT '8D' = UPDATE '8E' = DELETE
'12'	**PERFORM TRANSACTION OPERATION** P2 coding and meaning: '80' = BEGIN '81' = COMMIT '82' = ROLLBACK
'14'	**PERFORM USER OPERATION** P2 coding and meaning: '80' = PRESENT USER '81' = CREATE USER '82' = DELETE USER

The usage of these commands and encoding examples are shown in annex A.

6.3 Notation and special codings

In subsequent chapters the following notation is used for describing SQL statements:

— words in capital letters are SQL words (fixed expressions of the SQL language)

— [ ] means optional

— <...> means attribute string

— ::= means consists of

— | means or

— * means all

For encoding of parameters, the following notation is used:

— Lp = Length (coded in one byte) of the subsequent parameter

— <...> = parameter string of bytes with the length Lp and the meaning given in <...>

For encoding of a dimension D (e.g. no. of columns or no. of conditions), the following rule applies:

D ::= N
with N = no. of subsequent items, coded on one byte

or

D ::= Ln<N>
with Ln = '01' (N coded in one byte).

An item consists of one or several consecutive parameters. The null dimension is coded on one byte set at '00'. The meaning of a null dimension is either 'all columns' or 'no conditions' according to the command.

For the comparison operators which occur in search conditions, the coding according table 3 is used.

Table 3 — Coding of comparison operators

Comparison operator	Coding	Meaning
=	'3D'	equal to
<	'3C'	less than
>	'3E'	greater than
≤	'4C'	less than or equal to
≥	'47'	greater than or equal to
≠	'23'	not equal to

6.4 Status bytes

The status bytes SW1-SW2 of a response denote the processing state in the card. Table 4 shows the general meaning of the values of SW1-SW2 defined in this part of ISO/IEC 7816. For each command or performed operation, an appropriate clause provides more detailed meanings.

The meaning of status bytes defined in part 4 of this standard and listed here are defined more precisely for the usage of this part of the standard.

Table 4 — Status bytes

SW1-SW2	Defined in part	Meaning
		Normal processing
'9000'	4	Command successful
'61xx'	4	Command successful, xx codes the number of data bytes to be fetched by GET RESPONSE
		Warning processing
'6282'	4	End of table reached
		Execution errors
'6500'	4	No information given
'6581'	4	Memory failure (e.g. info corrupted)
		Checking errors
'6700'	4	Wrong length
		Command not allowed
'6900'	4	No information given
'6982'	4	Security status not satisfied
'6985'	4	Necessary commands or operations not performed before
		Wrong parameters
'6A00'	4	No information given
'6A80'	4	Incorrect parameter in data field
'6A81'	4	Operation not supported
'6A84'	4	Not enough memory space
'6A88'	4	Referenced object not found
'6A89'	7	Object exists already
'6Cxx'	4	Wrong length Le: SW2 indicates the exact length
'6D00'	4	Instruction code not supported

6.5 Coding of identifiers

The following conventions for identifiers are defined :

<identifier> ::= <capital letter> [<capital letter> | <digit> | <_>]
<capital letter> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<table name> ::= <identifier, max. 8 bytes>
<view name> ::= <identifier, max. 8 bytes>
<dictionary name> ::= <specifiable part of dictionary name><_><O|U|P>
<column name> ::= <identifier, max. 8 bytes>
<specifiable part of dictionary name> ::= <identifier, max. 6 bytes> | SYSTAB
<user id> ::=
    <individual id> |
    <group id> <delimiter> <individual id> |
    <group id> <delimiter> <subgroup id>
    <delimiter> <individual id> |
    <group id> <delimiter> <asterisk> |
    <group id> <delimiter> <subgroup id>
    <delimiter> <asterisk> |
    <group id> <delimiter> <asterisk>
    <delimiter> <asterisk>

<group id> ::= <identifier, max. 8 bytes>
<subgroup id> ::= <identifier, max. 8 bytes>
<individual id> ::= <identifier, max. 8 bytes> | <special user id>

<delimiter> ::= .
<asterisk> ::= *

<special user id> ::= <cardholder> | <public user>
<cardholder> ::= CHOLDER
<public user> ::= PUBLIC

CHOLDER is the general user id for the cardholder, PUBLIC is the general user id for a database basic user, see table 1.

The meaning of an asterisk is 'don't care', i.e. the coding of this part is not compared.

For checking a user id, the following cases have to be distinguished:

If the user id is an individual id, then the user id has to be identical with the registered user id.

If the user id consists of a group id in combination with an individual id, then the following steps have to be performed:

1) check whether the full user id is registered

2) if not, check, whether <group id>.* is registered

If the user id consists of a group id in combination with subgroup id and individual id, then the following steps have to be performed:

a) check whether the full user id is registered

b) if not, check,whether <group id>.<subgroup id>.* is registered

c) if not, check, whether <group id>.*.* is registered

The user id verification is performed, if a PRESENT USER operation is performed, but also in situations where access control to tables, views and dictionaries is required (see DECLARE CURSOR and INSERT).

NOTE The user group construction mechanism is not part of SQL.

6.6 Security attributes of tables, views and users

The following conventions for security attributes are defined :

<security attribute> ::= <security related data object as defined in other parts of this standard, e.g. for authentication or access control>

Security attributes associated to tables and views may be related to authentication procedures to be performed before access or describe secure messaging mechanisms to be applied, if data manipulation operations are performed (e.g. reading and writing in a confidential mode).

A security attribute attached to a user is related to user authentication.

6.7 Linking user ids to INSERT and UPDATE operations

If the last column of a table has the name USER, then the card will maintain a record of the user making the last modification to the table. The linking operation consists of inserting the current user id, set with the PRESENT USER operation, in the column USER when performing the INSERT operation. When performing the UPDATE operation, then the card overwrites the existing user id in the column USER with the current user id.

7 Database operations

7.1 CREATE TABLE

7.1.1 Definition and scope

The SCQL operation CREATE TABLE defines a table with its columns and possibly with security attributes. The table definition is added in the object description table.

7.1.2 Conditional usage and security

A table can only be created by users with the profile DB_O and DBOO.

7.1.3 Command message

The SCQL operation is related to the following SQL statement:

CREATE TABLE <table name> <table element list> [<security attribute>, ...]
<table name> ::= <identifier, see 6.5>
<table element list> ::= (<column definition> [, <column definition> ...] [<columnUSER>])
<security attribute> ::= <security related DO, see 6.6>

<column definition> ::= <column name>
[<delimiter><unique constraint definition>]
[<delimiter><data type>]

<column name> ::= <identifier, see 6.5>
<columnUSER, see 6.7> ::= USER

<unique constraint definition> ::= U
<delimiter> ::= .
<data type> ::= <variable character (length)>

<variable character (length)> ::= V<length>
<length> ::= <binary coded length on 1 byte>

If the unique constraint definition is used and supported, then the card has to ensure that all values in the related column are unique.

If the variable length indication (i.e. maximum length) is present and supported, then the card has to check that the presented length of the column does not exceed the specified maximum length.

Table 5 — PERFORM SCQL OPERATION command APDU for CREATE TABLE

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'80' = CREATE TABLE
Lc field	Length of subsequent data field
Data field	Lp <table name, see 6.5> D, fixing N (columns) N items: Lp <column definition> Optional parameters: Lp <max. no. of rows, binary coded on 1 byte> Lp <security attribute> [<security attribute>, ...]
Le field	Empty

7.1.4 Response message

Table 6 — PERFORM SCQL OPERATION response APDU for CREATE TABLE

Data field	Empty
SW1-SW2	Status bytes

7.1.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '84': Not enough memory space
- '89': Object exists already

7.2 CREATE VIEW

7.2.1 Definition and scope

The SCQL operation CREATE VIEW defines a view on a table. The view definition is added in the object description table.

7.2.2 Conditional usage and security

A view can only be created by the owner of the referenced table.

7.2.3 Command message

The SCQL operation is related to the following SQL statement:

CREATE VIEW <view name> AS <view definition> [<security attribute>, ...]
<view name> = <identifier, see 6.5>

<view definition> ::= SELECT <select list> FROM <object name> [WHERE <search condition> [AND <search condition>, ...]]
<security attribute>::= <security related DO, see 6.6>
<select list> ::= * | <column name> [, <column name>]
<object name> ::= <table name>
<search condition> ::= <column name> <comparison operator> <string>
<comparison operator> ::= = | < | > | ≤ | ≥ | ≠
<string> ::= '<sequence of bytes>'
* = all columns

Table 7 — PERFORM SCQL OPERATION command APDU for CREATE VIEW

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'81' = CREATE VIEW
Lc field	Length of subsequent data field
Data field	Lp <view name, see 6.5> Lp <table name> D, fixing N (columns) N items: Lp <column name> D, fixing N (conditions) N items consisting of 3 parameters: Lp <column name> Lp <comparison operator> Lp <string> Optional parameters: Lp <security attribute> [<security attribute>, ...]
Le field	Empty

NOTE If several conditions are present, they are implicitly combined with a logical AND.

7.2.4 Response message

Table 8 — PERFORM SCQL OPERATION response APDU for CREATE VIEW

Data field	Empty
SW1-SW2	Status bytes

7.2.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '84': Not enough memory space
- '88': Referenced object not found
- '89': Object exists already

7.3 CREATE DICTIONARY

7.3.1 Definition and scope

The SCQL operation CREATE DICTIONARY defines a view on the system tables *O, *U and *P. The fixed view definitions are added by the card in the object description table, see tables 11 and 12. The rows of the system tables which can be read, are dependent of the user's profile.

NOTE This command has no equivalence in SQL.

7.3.2 Conditional usage and security

A dictionary can only be created by the DB_O or a DBOO.

7.3.3 Command message

The SCQL operation is related to the following SQL extension statement:

CREATE DICTIONARY <specifiable part of the dictionary name>

<specifiable part of the dictionary name> ::= <identifier, max. 6 bytes, see 6.5> | SYSTAB

NOTE The specifiable part of the dictionary name is completed by the card by adding _O for the view of the object description table, _U for the view of the user description table and _P for the view of the privilege description table. As a general dictionary name, if needed, SYSTAB shall be used.

Table 9 — PERFORM SCQL OPERATION command APDU for CREATE DICTIONARY

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'82' = CREATE DICTIONARY
Lc field	Length of subsequent data field
Data field	Lp <specifiable part of the dictionary name, see 6.5>
Le field	Empty

Table 10 — Rows inserted, if user is DB_O

*O (Object description table)			
OBJNAM	**OBJOWN**	**OBJTYP**	**OBJDES**
<specifiable dictionary name>_O	<user id of DB_O>	V	ALL or name of columns
<specifiable dictionary name>_U	<user id of DB_O>	V	ALL or name of columns
<specifiable dictionary name>_P	<user id of DB_O>	V	ALL or name of columns

NOTE In table 10 the column OBJOPT is empty and not shown.

Table 11 — Rows inserted, if user is DBOO

*O (Object description table)			
OBJNAM	**OBJOWN**	**OBJTYP**	**OBJDES**
<specifiable dictionary name>_O	<user id of DBOO>	V	ALL or name of columns; condition: OBJOWN = <user id of DBOO>
<specifiable dictionary name>_U	<user id of DBOO>	V	ALL or name of columns; condition: USROWN = <user id of DBOO>
<specifiable dictionary name>_P	<user id of DBOO>	V	ALL or name of columns; condition: OBJOWN = <user id of DBOO>

NOTE In table 11 the column OBJOPT is empty and not shown.

7.3.4 Response message

Table 12 — PERFORM SCQL OPERATION response APDU for CREATE DICTIONARY

Data field	Empty
SW1-SW2	Status bytes

7.3.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '84': Not enough memory space
- '89': Object exists already

7.4 DROP TABLE

7.4.1 Definition and scope

With the SCQL operation DROP TABLE a table can be dropped.

7.4.2 Conditional usage and security

A table can only be dropped by its owner. The privileges associated to the table should be automatically dropped.

7.4.3 Command message

The SCQL operation is related to the following SQL statement:

DROP TABLE <table name>

Table 13 — PERFORM SCQL OPERATION command APDU for DROP TABLE

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'83' = DROP TABLE
Lc field	Length of subsequent data field
Data field	Lp <table name>
Le field	Empty

7.4.4 Response message

Table 14 — PERFORM SCQL OPERATION response APDU for DROP TABLE

Data field	Empty
SW1-SW2	Status bytes

7.4.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '88': Referenced object not found

7.5 DROP VIEW

7.5.1 Definition and scope

With the SCQL operation DROP VIEW a view can be dropped.

7.5.2 Conditional usage and security

A view can only be dropped by its owner. The privileges associated to the view should be automatically dropped.

7.5.3 Command message

The SCQL operation is related to the following SQL statement:

DROP VIEW <view name or dictionary name >

Table 15 — PERFORM SCQL OPERATION command APDU for DROP VIEW

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'84' = DROP VIEW
Lc field	Length of subsequent data field
Data field	Lp <view name or dictionary name>
Le field	Empty

7.5.4 Response message

Table 16 — PERFORM SCQL OPERATION response APDU for DROP VIEW

Data field	Empty
SW1-SW2	Status bytes

7.5.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '88': Referenced object not found

7.6 GRANT

7.6.1 Definition and scope

The SCQL operation GRANT allows to grant privileges to a single user, to a user group or to all users.

The following privileges may be granted:

a) Privileges for table access

— SELECT

— INSERT

— UPDATE

— DELETE

b) Privileges for view access

— SELECT

— UPDATE

c) Privileges for dictionary access

— SELECT.

NOTE If in addition to a privilege an access authorization by the cardholder shall be required (i.e. password presentation) before the respective action can be performed, then this has to be defined in the security attributes defined for the respective table or view.

7.6.2 Conditional usage and security

Only the owner of the table or view can grant or revoke privileges.

7.6.3 Command message

The SCQL operation is related to the following SQL statement:

GRANT <privileges> ON <object name> TO <grantee>

<privileges> ::= <action> [,<action> ...] | ALL
<action> ::= SELECT | INSERT | UPDATE | DELETE

<object name> ::= <table name> | <view name> | <dictionary name>

<grantee> ::= <user id, see 6.5> | *
* = all users

Table 17 — PERFORM SCQL OPERATION command APDU for GRANT

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'85' = GRANT
Lc field	Length of subsequent data field
Data field	Lp <privileges, coding see table 18> Lp <table name, view name or dictionary name> Lp <user id (see 6.5) or *>
Le field	Empty

Table 18 — Coding of privileges

Privilege	Coding in SCQL
INSERT	'41'
SELECT	'42'
UPDATE	'44'
DELETE	'48'
all	'4F'

7.6.4 Response message

Table 19 — PERFORM SCQL OPERATION response APDU for GRANT

Data field	Empty
SW1-SW2	Status bytes

7.6.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '84': Not enough memory space
- '88': Referenced object not found

7.7 REVOKE

7.7.1 Definition and scope

The SCQL operation REVOKE allows to revoke privileges granted before (see 7.6).

7.7.2 Conditional usage and security

Only the owner of the table or view can revoke privileges.

7.7.3 Command message

The SCQL operation is related to the following SQL statement:

REVOKE <privileges> ON <object name> FROM <grantee>

<privileges> ::= <action> [,<action> ...] | ALL
<action> ::= SELECT | INSERT | UPDATE | DELETE

<object name> ::= <table name> | <view name> | <dictionary name>

<grantee> ::= <user id> | *
* = all users

Table 20 — PERFORM SCQL OPERATION command APDU for REVOKE

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'86' = REVOKE
Lc field	Length of subsequent data field
Data field	Lp <privileges, coding see table 18> Lp <table name, view name or dictionary name> Lp <user id or *>
Le field	Empty

7.7.4 Response message

Table 21 — PERFORM SCQL OPERATION response APDU for REVOKE

Data field	Empty
SW1-SW2	Status bytes

7.7.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '88': Referenced object not found

7.8 DECLARE CURSOR

7.8.1 Definition and scope

A cursor is used for pointing to a row in a table, view or dictionary. The SCQL operation DECLARE CURSOR is used for the declaration of a cursor.

7.8.2 Conditional usage and security

The declaration of the cursor is only accepted, if the actual user is authorized to access the referenced table, view or dictionary. The user has to be the owner of the referenced object or at least one privilege for access to the referenced object (for comparison of the current user id with the user id stored in the system table *P see 6.5).

Only one cursor can exist at a given time, i.e. if a new cursor is declared then the previous is no longer valid.

7.8.3 Command message

The SCQL operation is related to the following SQL statement:

DECLARE CURSOR FOR <selection>

<selection> ::= SELECT <select list> FROM <object name> [WHERE <search condition> [AND <search condition>, ...]

<select list> ::= * | <column name> [, <column name>]

<object name> ::= <table name> | <view name> | <dictionary name>

<search condition> ::= <column name> <comparison operator> <string>

<comparison operator> ::= = | < | > | ≤ | ≥ | ≠

<string> ::= '<sequence of bytes>'

* = all columns

NOTE Since only one cursor at a time is possible, no cursor name is used.

Table 22 — PERFORM SCQL OPERATION command APDU for DECLARE CURSOR

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'87' = DECLARE CURSOR
Lc field	Length of subsequent data field
Data field	Lp <table name, view name or dictionary name> D, fixing N (columns) N items: Lp <column name> If conditions are present: D, fixing N (conditions) N items consisting of 3 parameters: Lp <column name> Lp <comparison operator> Lp <string>
Le field	Empty

NOTE If several conditions are present, they are implicitly combined with a logical AND.

7.8.4 Response message

Table 23 — PERFORM SCQL OPERATION response APDU for DECLARE CURSOR

Data field	Empty
SW1-SW2	Status bytes

7.8.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in data field
- '88': Referenced object not found

7.9 OPEN

7.9.1 Definition and scope

The SCQL operation OPEN opens a cursor, i.e. the cursor is positioned on the first row which satisfies the selection previously defined with the DECLARE CURSOR operation.

7.9.2 Conditional usage and security

A cursor must be declared before.

7.9.3 Command message

The SCQL operation is related to the following SQL statement:

OPEN

Table 24 — PERFORM SCQL OPERATION command APDU for OPEN

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'88' = OPEN
Lc field	Empty
Data field	Empty
Le field	Empty

7.9.4 Response message

Table 25 — PERFORM SCQL OPERATION response APDU for OPEN

Data field	Empty
SW1-SW2	Status bytes

7.9.5 Status conditions

The following specific error conditions may occur:

— SW1 = '62' with SW2 =

- '82': End of table reached

— SW1 = '69' with SW2 =

- '85': Necessary commands or operations not performed before (no cursor defined)

7.10 NEXT

7.10.1 Definition and scope

The SCQL operation NEXT sets the cursor on the next row satisfying the cursor specification.

7.10.2 Conditional usage and security

A cursor must be opened before.

7.10.3 Command message

The SCQL operation is related to the following SQL statement:

NEXT

Table 26 — PERFORM SCQL OPERATION command APDU for NEXT

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'89' = NEXT
Lc field	Empty
Data field	Empty
Le field	Empty

7.10.4 Response message

Table 27 — PERFORM SCQL OPERATION response APDU for NEXT

Data field	Empty
SW1-SW2	Status bytes

7.10.5 Status conditions

The following specific error conditions may occur:

— SW1 = '62' with SW2 =

- '82': End of table reached

— SW1 = '69' with SW2 =

- '85': Necessary commands or operations not performed before (no cursor defined)

7.11 FETCH

7.11.1 Definition and scope

The SCQL operation FETCH allows to fetch a row or part of it. The cursor has to point on the row to be fetched.

7.11.2 Conditional usage and security

The operation can only be executed by the object owner or a user with the SELECT privilege. A cursor must be opened before.

7.11.3 Command message

The SCQL operation is related to the following SQL statement:

FETCH

Table 28 — PERFORM SCQL OPERATION command APDU for FETCH

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'8A' = FETCH
Lc field	Empty
Data field	Empty
Le field	Max. length of expected data

7.11.4 Response message

Table 29 — PERFORM SCQL OPERATION response APDU for FETCH

Data field	D, fixing N (columns)
SW1-SW2	N items: Lp <string> Status bytes

NOTE In the case the T=0 transmission protocol is used, the length of the data selected is indicated in the status bytes (SW1-SW2 = '6Cxx', where xx indicates the number of data bytes available). The data shall be retrieved by re-issuing the same command with the value of Le indicated in SW2.

In the case the T=1 transmission protocol is used, the data are transmitted in the FETCH response APDU.

7.11.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied
- '85': Necessary commands or operations not performed before (no cursor defined)

7.12 FETCH NEXT

7.12.1 Definition and scope

The SCQL operation FETCH NEXT has to be used for reading the logical next row from the cursor position. The cursor is set to the row fetched.

7.12.2 Conditional usage and security

The operation can only be executed by the object owner or a user with the SELECT privilege. A cursor must be opened before.

7.12.3 Command message

The SCQL operation is related to the following SQL statement:

FETCH NEXT

Table 30 — PERFORM SCQL OPERATION command APDU for FETCH NEXT

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'8B' = FETCH NEXT
Lc field	Empty
Data field	Empty
Le field	Max. length of expected data

7.12.4 Response message

Table 31 — PERFORM SCQL OPERATION response APDU for FETCH NEXT

Data field	D, fixing N (columns)
SW1-SW2	N items: Lp <string> Status bytes

See note below table 29.

7.12.5 Status conditions

The following specific error conditions may occur:

— SW1 = '62' with SW2 =

- '82': End of table reached

— SW1 = '69' with SW2 =

- '82': Security status not satisfied
- '85': Necessary commands or operations not performed before (no cursor defined)

7.13 INSERT

7.13.1 Definition and scope

The SCQL operation INSERT is used to insert a row in a table. A new row is always added at the end of a table. The cursor remains at its position.

7.13.2 Conditional usage and security

The command can only be executed by the table owner or a user with the INSERT privilege.

The value for the special coulmn USER – if present – is inserted by the card, see 6.7.

7.13.3 Command message

The SCQL operation is related to the following SQL statement:

INSERT [INTO] <table name> VALUES (<string> [,<string> ...])

<string> ::= '<sequence of bytes>'

Table 32 — PERFORM SCQL OPERATION command APDU for INSERT

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'8C' = INSERT
Lc field	Length of subsequent data field
Data field	Lp <table name> D, fixing N (columns) N items: Lp <string>
Le field	Empty

7.13.4 Response message

Table 33 — PERFORM SCQL OPERATION response APDU for INSERT

Data field	Empty
SW1-SW2	Status bytes

7.13.5 Status conditions

The following specific error conditions may occur:

— SW1 = '62' with SW2 =

- '82': End of table reached

— SW1 = '67' with SW2 =

- '00': Wrong length

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in the data field
- '84': Not enough memory space
- '88': Referenced object not found
- '89': Object exists already (column value not unique)

7.14 UPDATE

7.14.1 Definition and scope

The SCQL operation UPDATE updates one or more fields of a row in a table or view to which the cursor points.

7.14.2 Conditional usage and security

The command can only be executed by the table owner or a user with the UPDATE privilege. A cursor must be opened before.

The value for the special column USER – if present – is modified by the card, see 6.7.

7.14.3 Command message

The SCQL operation is related to the following SQL statement:

UPDATE SET <set clause list>

<set clause list> ::= <column name> = <string> [,<column name> = <string>...]

<string> ::= '<sequence of bytes>'

Table 34 — PERFORM SCQL OPERATION command APDU for UPDATE

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'8D' = UPDATE
Lc field	Length of subsequent data field
Data field	D, fixing N (columns) N items consisting of 2 parameters: Lp <column name> Lp <string>
Le field	Empty

7.14.4 Response message

Table 35 — PERFORM SCQL OPERATION response APDU for UPDATE

Data field	Empty
SW1-SW2	Status bytes

7.14.5 Status conditions

The following specific error conditions may occur:

— SW1 = '67' with SW2 =

- '00': Wrong length

— SW1 = '69' with SW2 =

- '82': Security status not satisfied
- '85': Necessary commands or operations not performed before (no cursor defined)

— SW1 = '6A' with SW2 =

- '80': Incorrect parameter(s) in the data field
- '84': Not enough memory space
- '89': Object exists already (column value not unique)

7.15 DELETE

7.15.1 Definition and scope

With the SCQL operation DELETE a row in a table to which the cursor points, can be deleted. The cursor is moved to the logical next row.

7.15.2 Conditional usage and security

The command can only be executed by the table owner or a user with the DELETE privilege for the referenced table.

7.15.3 Command message

The SCQL operation is related to the following SQL statement:

DELETE

Table 36 — PERFORM SCQL OPERATION command APDU for DELETE

CLA	As defined in ISO/IEC 7816-4
INS	'10' (= PERFORM SCQL OPERATION)
P1	'00', other values RFU
P2	'8E' = DELETE
Lc field	Empty
Data field	Empty
Le field	Empty

7.15.4 Response message

Table 37 — PERFORM SCQL OPERATION response APDU for DELETE

Data field	Empty
SW1-SW2	Status bytes

7.15.5 Status conditions

The following specific error conditions may occur:

— SW1 = '62' with SW2 =

- '82': End of table reached

— SW1 = '69' with SW2 =

- '82': Security status not satisfied
- '85': Necessary commands or operations not performed before (no cursor defined)

8 Transaction management

8.1 General concept

A transaction in this context is the process of a modification in the database. A modification can be an update or an insertion of one or more rows.

The PERFORM TRANSACTION OPERATION command provides the operations needed for confirmation or cancellation of transactions.

If this command is not used or not supported, an SCQL operation is always executed in the sense of commit, i.e. a modification caused by the operation becomes immediately valid.

8.2 Transaction operations

8.2.1 BEGIN

8.2.1.1 Definition and scope

The transaction operation BEGIN allocates space for a memory image, e.g. a row.

8.2.1.2 Conditional usage and security

The memory space which is provided is implementation dependent. It is recommended that enough memory space for the buffering of at least one row is allocated.

8.2.1.3 Command message

Table 38 — PERFORM TRANSACTION OPERATION command APDU for BEGIN

CLA	As defined in ISO/IEC 7816-4
INS	'12' (= PERFORM TRANSACTION OPERATION)
P1	'00', other values RFU
P2	'80' = BEGIN
Lc field	Empty
Data field	Empty
Le field	Empty

8.2.1.4 Response message

Table 39 — PERFORM TRANSACTION OPERATION response APDU for BEGIN

Data field	Empty
SW1-SW2	Status bytes

8.2.1.5 Status conditions

The following specific error conditions may occur:

— SW1 = '6A' with SW2 =

- '84' : Not enough memory space

8.2.2 COMMIT

8.2.2.1 Definition and scope

The transaction operation COMMIT validates all the modifications made since the transaction operation BEGIN has been executed.

8.2.2.2 Conditional usage and security

The transaction operation BEGIN has to be previously performed.

8.2.2.3 Command message

Table 40 — PERFORM TRANSACTION OPERATION command APDU for COMMIT

CLA	As defined in ISO/IEC 7816-4
INS	'12' (= PERFORM TRANSACTION OPERATION)
P1	'00', other values RFU
P2	'81' = COMMIT
Lc field	Empty
Data field	Empty
Le field	Empty

8.2.2.4 Response message

Table 41 — PERFORM TRANSACTION OPERATION response APDU for COMMIT

Data field	Empty
SW1-SW2	Status bytes

8.2.2.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '85' : Necessary commands or operations not performed before (no BEGIN transaction)

8.2.3 ROLLBACK

8.2.3.1 Definition and scope

The transaction operation ROLLBACK restores the context in the way it was before the transaction operation BEGIN has been executed.

8.2.3.2 Conditional usage and security

The transaction operation BEGIN has to be previously performed.

8.2.3.3 Command message

Table 42 — PERFORM TRANSACTION OPERATION command APDU for ROLLBACK

CLA	As defined in ISO/IEC 7816-4
INS	'12' (= PERFORM TRANSACTION OPERATION)
P1	'00', other values RFU
P2	'82' = ROLLBACK
Lc field	Empty
Data field	Empty
Le field	Empty

8.2.3.4 Response message

Table 43 — PERFORM TRANSACTION OPERATION response APDU for ROLLBACK

Data field	Empty
SW1-SW2	Status bytes

8.2.3.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '85' : Necessary commands or operations not performed before (no BEGIN transaction)

9 User management

9.1 General concept

User management is related to

— user identification

— user authentication

— user authorization

— user registration/deregistration.

For user identification a user id is used, which may be structured as outlined in 6.5.

If there is the need for a proof that the presented user id is associated to the user, an authentication procedure has to be performed. Appropriate mechanisms for user authentication are e.g.

— a password verification mechanism

— a biometrical verification mechanism

— a cryptographic mechanism based e.g. on a symmetric algorithm or a public key system in combination with certificates.

In a certificate a user id suitable for user identification may be present.

The user authorization deals with the user's rights to perform operations and actions. In the SCQL environment the authorization of a user is linked to

— the user profile (DB_O, DBOO or DBBU),

— the user privileges (see tab. 18) and optionally

— the user role respectively the user group (see 6.5, 7.6, 7.7 and 9.2.1).

The user registration covers the registration of a user with its user id, its user profile and possibly with its security attributes (see 6.6). The user deregistration is performed by deletion of the registration.

In this clause, the following operations for user identification based on the PERFORM USER OPERATION command are specified:

— PRESENT USER

— CREATE USER

— DELETE USER.

Commands and operations related to an authentication procedure are not specified in this part of the standard. However, when creating a user with its user id and its user profile, security attributes may be set which consists of suitable security related DOs defined in other parts of the standard. Special commands and operations related to authorization can be found in clause 7 (see CREATE VIEW, GRANT and REVOKE).

9.2 User operations

9.2.1 PRESENT USER

9.2.1.1 Definition and scope

With the PRESENT USER operation, the registration of the presented user id is checked. The user id shall be presented according to the conventions defined in clause 6.5 or in the cardholder name DO present in the cardholder's certificate. If the user id is registered in the system table *U, the user characterized by its user id is set as current user.

NOTE A certificate containing a cardholder name DO may belong e.g. to a professional using his professional card or to a service provider.

9.2.1.2 Conditional usage and security

There can be only one current user at a time per logical channel.

As subsequent commands authentication commands may follow e.g.

— VERIFY or

— GET CHALLENGE and EXTERNAL AUTHENTICATE, see ISO/IEC 7816-4.

9.2.1.3 Command message

The PRESENT USER operation is related to the following SQL extension statement:

PRESENT USER <user id, see 6.5> | <cardholder certificate DO>

<cardholder certificate DO> ::= <tag '7F21'><length> <cardholder name DO> <additional DOs>

<cardholder name DO> ::= <tag '5F20'>
<length> <user id, see 6.5>

Table 44 — PERFORM USER OPERATION command APDU for PRESENT USER

CLA	As defined in ISO/IEC 7816-4
INS	'14' (= PERFORM USER OPERATION)
P1	'00', other values RFU
P2	'80' = PRESENT USER
Lc field	Length of subsequent data field
Data field	<user id, see 6.5> or <cardholder certificate DO, tag '7F21', see ISO/IEC 7816-6>
Le field	Empty

9.2.1.4 Response message

Table 45 — PERFORM USER OPERATION response APDU for PRESENT USER

Data field	Empty
SW1-SW2	Status bytes

9.2.1.5 Status conditions

The following specific error conditions may occur:

— SW1 = '6A' with SW2 =

- '80' : Incorrect parameter(s) in data field
- '88' : Referenced object (user id) not found

9.2.2 CREATE USER

9.2.2.1 Definition and scope

The CREATE USER operation initiates the registration of a user. In an SCQL environment a row in the user description table is inserted by the card.

If authentication is required, when the user wants to access the database or protected tables or views, authentication related information has to be added (see 6.6). In this case the CREATE USER operation may be followed by a command e.g. for installing a password. The specification of those commands is outside the scope of this part of the standard.

9.2.2.2 Conditional usage and security

The CREATE USER command can only be performed by users with profile DB_O or DBOO with the permissions described in table 1. The user id has to be unique.

9.2.2.3 Command message

The CREATE USER operation is related to the following SQL extension statement:

CREATE USER <user id> <user profile> [<security attribute>, ...]

<user profile> ::= <database object owner> | <database basic user>

<database object owner> ::= DBOO
<database basic user> ::= DBBU

<security attribute>::= <security related DO, see 6.6>

Table 46 — PERFORM USER OPERATION command APDU for CREATE USER

CLA	As defined in ISO/IEC 7816-4
INS	'14' (= PERFORM USER OPERATION)
P1	'00', other values RFU
P2	'81' = CREATE USER
Lc field	Length of subsequent data field
Data field	Lp <user id, see 6.5> Lp <user profile: DBOO or DBBU> Optional parameters: Lp <security attribute, see 6.6>
Le field	Empty

NOTE A user with the profile DB_O can only be inserted in the user description table during the installation phase of the SCQL database.

9.2.2.4 Response message

Table 47 — PERFORM USER OPERATION response APDU for CREATE USER

Data field	Empty
SW1-SW2	Status bytes

9.2.2.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80' : Incorrect parameter(s) in data field
- '89' : Object (user id) exists already

9.2.3 DELETE USER

9.2.3.1 Definition and scope

With the DELETE USER operation, a user id can be deleted. The respective row in the user description table is erased.

9.2.3.2 Conditional usage and security

The DELETE USER operation can only be performed by the user owner.

In order to ensure database integrity privileges associated to this user should be automatically removed.

NOTE An asterisk in the user id presented with the DELETE USER operation (see 6.5) has in this case no special meaning, i.e. only the row in *U is deleted, where the presented user id is identical with the registered user id.

9.2.3.3 Command message

The DELETE USER operation is related to the following SQL extension statement:

DELETE USER <user id>

Table 48 — PERFORM USER OPERATION command APDU for DELETE USER

CLA	As defined in ISO/IEC 7816-4
INS	'14' (= PERFORM USER OPERATION)
P1	'00', other values RFU
P2	'82' = DELETE USER
Lc field	Length of subsequent data field
Data field	Lp <user id>
Le field	Empty

9.2.3.4 Response message

Table 49 — PERFORM USER OPERATION response APDU for DELETE USER

Data field	Empty
SW1-SW2	Status bytes

9.2.3.5 Status conditions

The following specific error conditions may occur:

— SW1 = '69' with SW2 =

- '82': Security status not satisfied

— SW1 = '6A' with SW2 =

- '80' : Incorrect parameter(s) in data field
- '88' : Referenced object (user id) not found

Annex A
(informative)

Usage of SCQL operations

The subsequent example shows the usage and the encoding of the PRESENT USER operation and some SCQL operations, when coding the dimension D on one byte.

The following abbreviations are used:

CH = command header (= CLA INS P1 P2)
col = column name
coldef = column definition
comp = comparison operator
tab = table name
view = view name
x = hexadecimal

PRESENT USER 'COMPANY.DIV.SMITH'

CH	Lc	user id
x00140080	x11	COMPANY.DIV.SMITH

CREATE TABLE FLY ('DEP', 'ARR', 'F_NO.U', 'TIME', 'PRICE')

CH	Lc	Lp	tab	N	Lp	col	Lp	col	Lp	coldef	Lp	col	Lp	col
x00100080	x1F	x03	FLY	x05	x03	DEP	x03	ARR	x06	F_NO.U	x04	TIME	x05	PRICE

NOTE F_NO.U means that the value in the column F_NO has to be unique

CREATE VIEW FLY_A AS SELECT ('DEP', 'ARR', 'F_NO', 'TIME') FROM FLY

CH	Lc	Lp	view	Lp	tab	N	Lp	col	Lp	col	Lp	col	Lp	col
x00100081	x1D	x05	FLY_A	x03	FLY	x04	x03	DEP	x03	ARR	x04	F_NO	x04	TIME

GRANT SELECT ON 'FLY_A' TO *

CH	Lc	Lp	Priv	Lp	view	Lp	user id
x00100085	x0A	x01	x42	x05	FLY_A	01	*

INSERT INTO 'FLY' VALUES ('FRA','CDG','LH4711','0115_10:20','540DM')

CH	Lc	Lp	tab	N	Lp	DEP	Lp	ARR	Lp	F_NO	Lp	TIME	Lp	PRICE
x0010008C	x25	x03	FLY	x05	x03	FRA	x03	CDG	x06	LH4711	x0A	0115_10:20	x05	540DM

DECLARE CURSOR FOR SELECT * FROM 'FLY' WHERE 'ARR' = 'CDG'

CH	Lc	Lp	tab	N	N	Lp	col	Lp	comp	Lp	string
x00100087	x10	x03	FLY	x00	x01	x03	ARR	x01	x3D	x03	CDG

NOTE The comparison operator '3D' means 'equal to'.

ICS 35.240.15

Price based on 36 pages