



ISO/IEC JTC 1/SC 32 N **0019**

Date: 1997-11-10

CD 13238-2

ISO/IEC JTC 1/SC 32/WG 3 (N2116)

Secretariat: SCC-Canada

6 **Information Technology — Data Management Export/Import — Part 2: SQL**
7 **Export/Import**

8 *Technologies de l'information — Échange (export/import) de données — Partie 1: Export/Import pour SQL*

Document type: International Standard

Document subtype: Not applicable

Document stage: (30) Committee

Document language: E

E:\berubJ\513-DBM\582-sq\OD_7-doc\32N0019_SQL_EI_323N2116_V10.doc
based on ISOSTD ISO Template Version 3.2 1997-07-10 Modif JBE 0.3 1997-09-15

1 Contents

2	1	Scope	1
3	2	Conformance	1
4	3	Normative reference(s)	2
5	4	Term(s) and definition(s)	3
6	4.1	From Other Standards	3
7	4.1.1	ISO/IEC 9075	3
8	4.1.2	ISO/IEC 8824-1	3
9	4.1.3	ISO/IEC 9075	3
10	4.1.4	ISO/IEC 15475-1	4
11	4.1.5	ISO/IEC 13238-1	4
12	4.2	For this standard	5
13	5	Symbols (and abbreviated terms)	6
14	5.1	Notation	6
15	5.1.1	ASN.1	6
16	5.1.1	BNF	7
17	5.2	Abbreviations	7
18	6	Concept and Facilities	7
19	6.1	Transfer File	7
20	6.1.1	Structure	7
21	6.1.2	Transfer file header	8
22	6.1.3	Transfer file SQL-definition component	8
23	6.1.4	Transfer file SQL-data component	8
24	6.2	Export/import processes	8
25	6.2.1	Export process	8
26	6.2.2	Import process	8
27	7	Specification of the transfer file	9

1	7.1	Codeset	9
2	7.1.1	Character Set.....	9
3	7.1.2	Character Set Encoding.....	9
4	7.2	Encoding	9
5	7.2.1	Export/import-binary-packed encodings	9
6	7.2.2	Export/import-clear-text encodings	10
7	8	Transfer file Structure.....	11
8	8.1	Syntax for transfer file	11
9	8.1.1	Content	11
10	8.1.2	Definition	11
11	8.1.3	General Rules.....	11
12	8.2	Transfer file header	11
13	8.2.1	Content	11
14	8.2.2	Definition	12
15	8.2.3	General Rules.....	13
16	8.3	Transfer file SQL-definition	14
17	8.3.1	Content	14
18	8.3.2	Definition	14
19	8.3.3	General Rules.....	14
20	8.4	Transfer file SQL-data.....	16
21	8.4.1	Content	16
22	8.4.2	Definition	16
23	8.4.3	General Rules.....	16
24	8.4.4	Syntax for transfer file SQL-data.....	17
25	8.4.5	Content	17
26	8.4.6	Definition	17
27	8.4.7	General Rules.....	17
28	9	Specification of export/import processes.....	19
29	9.1	Procedures provided	19

1	9.2	Selection of data to be exported	19
2	9.3	Open SQL Transfer file	20
3	9.3.1	Function	20
4	9.3.2	Definition	20
5	9.3.3	Input.....	20
6	9.3.4	Output.....	20
7	9.3.5	General Rules.....	21
8	9.4	Select For Export Procedure	21
9	9.4.1	Function	21
10	9.4.2	Format	21
11	9.4.3	Input.....	22
12	9.4.4	Output.....	22
13	9.4.5	General Rules.....	22
14	9.5	Clear Export Selection Procedure.....	22
15	9.5.1	Function	22
16	9.5.2	Format	22
17	9.5.3	Input.....	23
18	9.5.4	Output.....	23
19	9.5.5	General Rules.....	23
20	9.6	Export Procedure	23
21	9.6.1	Function	23
22	9.6.2	Definition	23
23	9.6.3	Input.....	24
24	9.6.4	Output.....	24
25	9.6.5	General Rules.....	24
26	9.6.6	Output from export process	25
27	9.7	SQLExportSelectedData	25
28	9.7.1	Function	25
29	9.7.2	Format	25

1	9.7.3	Input.....	26
2	9.7.4	Output.....	26
3	9.7.5	General Rules.....	26
4	9.8	Close SQL Transfer file Procedure.....	26
5	9.8.1	Function	26
6	9.8.2	Format	27
7	9.8.3	Input.....	27
8	9.8.4	Output.....	27
9	9.8.5	General Rules.....	27
10	9.9	Import Procedure	27
11	9.9.1	Function	27
12	9.9.2	Input parameters to import procedure	28
13	9.9.3	Output from import procedure	29
14	9.9.4	General Rules.....	30
15	10	Notes and Tables	33
16	Annex A (Informative)	Conventions mandated by use of ASN.1.....	34
17	Annex B (normative)	ASN.1 Definition of the Transfer Format	35
18	Annex C (informative)	Sample Transfer file Format.....	42

1 Foreword

2 *Boilerplate text for JTC 1 standards:*

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

9 International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

13 Editor's Foreword to the working draft (July 1995)

14 *N01- Editor's note This foreword will be removed in the final*
15 *document*

16 ISO/IEC JTC1/SC21/WG3 has developed a Standardization Framework for Data Management Export/Import (to be
17 IS 13238-1 (currently SC21 WG3 N1991). Parts 2 and 3 of this three-part standard are SQL Export/Import and
18 IRDS Export/Import. This attached working draft is for the SQL Export/Import.

19 This document has been sent out for review and comment as per recommendation xx of the WG3 Ottawa Plenary
20 (95.07.17).

1 Editor's Foreword to this committee draft (September 1997)

2 *N02- Editor's note This foreword will be removed in the final*
3 *document*

4 This document generally incorporates the following changes:

- 5 1 The changes recommended as a result of the circulation of the working draft.
- 6 2 The changes resulting from the impact of the progression of the two other parts (Framework and IRDS).
- 7 3 The changes recommended at the Madrid and London meetings.
- 8 4 The document has also been reformatted using ITTF –supplied templates.

9 The major changes to the document are:

- 10 1 Services for selecting the data to be exported have been added.
- 11 2 The specification style for services now used SQL PSM.
- 12 3 The ASN.1 code has been moved to a normative annex.
- 13 4 The definition of the syntax of the transfer file is done using BNF.
- 14 5 The file structure has been harmonized with the Framework, and the proposed structure for IRDS.
- 15 6 The definition clause has been harmonized with the definitions provided by the framework.

16 The other changes are:

- 17 1 The headings for defining the transfer file components have been improved and made consistent.
- 18 2 The definition of the transfer format for tables has been isolated in one subclause.
- 19 3 The name export/import file has been consistently replaced by transfer file in the document.
- 20 4 The same convention for capitalization and identification of tokens have been applied to the document. Tf
21 replaces eif, EIF, Eif, and Sql replaces SQL, sql and Sql.
- 22 5 The conformance clause has been moved to the beginning of the document, as per the ISO templates.

23

1 Introduction

2
3

*N03- Editor's note This will be replaced at the DIS stage by the
ITTF approved text*

4 The export of data from one computer system to another and the import of data into one computer system from
5 another are already a widespread practice. The first aim of this part of this International Standard is to provide a
6 standardized export/input file format which can be used for SQL-data, namely data which has been structured
7 according to Database Language SQL (see ISO/IEC 9075:1992).

8 The second aim is to define two generic applications program interfaces (API) which can be used for exporting data
9 from a computer system into an export/import file and for importing data from an export/input file into a computing
10 system. The physical means by which the export and the import take place is outside the scope of this International
11 Standard.

1 Information Technology — Data Management Export/Import — Part 2 2: SQL Export/Import

3 1 Scope

4 This part of ISO/IEC 13238 defines:

- 5 a) Two transfer file formats for the bulk transfer of SQL-data.
- 6 b) Two generic applications program interfaces, firstly for exporting SQL-data and secondly for importing SQL-
7 data.

8 The implementation of the selection mechanisms for the SQL-data and/or SQL-definition to be exported and/or
9 imported is outside the scope of this International Standard.

10 *N04- Modified because of the addition of the "Select for Export*
11 *Service". Maybe the sentence needs to be removed.* ||

12 2 Conformance

13 A system conforming to this International Standard can make one or more of the following conformance claims. If it
14 satisfies the requirements of all claims listed below, it is a fully conforming system. If it satisfies one or more it is a
15 partially conforming system. Claims for partial conformance shall state which of the following claims are satisfied.
16 The conformance claims are:

17 EXPORT-CLEAR TRANSFER FILE: The implementation is capable of generating an export/import-clear-text
18 encoding of a transfer file which conforms to the specification of clause 7.3 and the clauses that this references.

19 IMPORT-CLEAR TRANSFER FILE: The implementation is capable of processing an export/import-clear-text
20 encoding of a transfer file which conforms to the specification of clause 7.3 and the clauses that that references,
21 supporting all features of that encoding that are specified as an exporter's option.

22 EXPORT PACKED-TRANSFER-FILE: The implementation is capable of generating an export/import-binary-packed
23 encoding of a transfer file which conforms to the specification of clause 7.2 and the clauses that that references.

24 IMPORT CLEAR TRANSFER-FILE: The implementation is capable of processing an export/import-binary-packed
25 encoding of a transfer file which conforms to the specification of clause 7.2 and the clauses that that references,
26 supporting all features of that encoding that are specified as an exporter's option.

27 CLEAR-EXPORT-API: The implementation is capable of generating an export/import-clear-text encoding of a
28 transfer file which conforms to the specification of clause 7.3 and the clauses that that references, and provides a
29 documented interface which supports the passing of all the parameters specified in 9.1, and their use as specified
30 in that clause.

31 CLEAR-IMPORT-API: The implementation is capable of processing an export/import-clear-text encoding of a
32 transfer file which conforms to the specification of clause 7.3 and the clauses that this references, supporting all
33 features of that encoding that are specified as an exporter's option, and provides a documented interface which
34 supports the passing of all the parameters specified in 9.2, and their use as specified in that clause.

1 PACKED-EXPORT-API: The implementation is capable of generating an export/import-binary-packed encoding of
2 a transfer file which conforms to the specification of clause 7.2 and the clauses that that references, and provides a
3 documented interface which supports the passing of all the parameters specified in 9.1, and their use as specified
4 in that clause.

5 PACKED-IMPORT-API: The implementation is capable of processing an export/import-binary-packed encoding of a
6 transfer file which conforms to the specification of clause 7.2 and the clauses that that references, supporting all
7 features of that encoding that are specified as an exporter's option, and provides a documented interface which
8 supports the passing of all the parameters specified in 9.2, and their use as specified in that clause.

9 Note 1: Systems conforming for XXX-API automatically conform for XXX-FILE.

10 Note 2: Interworking is possible between a system that conforms for PACKED-EXPORT-FILE and one that conforms for
11 PACKED-IMPORT-FILE.

12 Note 4: Interworking is possible between a system that conforms for CLEAR-EXPORT-FILE and one that conforms for
13 CLEAR-IMPORT-FILE.

14 Note 5: Interworking is NOT possible between a system that conforms for PACKED-EXPORT-FILE and one that conforms for
15 CLEAR-IMPORT-FILE, unless it also supports other conformance claims.

16 *N05- Editor's note From an SQL point of view, it needs to be*
17 *considered whether it is possible to define a level of*
18 *conformance which is independent of ASN.1*

19 3 Normative reference(s)

20 The following International Standards contain provisions that, through reference in this text, constitute provisions of
21 this part of this International Standard. At the time of publication, the editions indicated were valid. All standards are
22 subject to revision, and parties to agreements based on this part of this part of ISO/IEC 13238 are encouraged to
23 investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC
24 and ISO maintain registers of currently valid International Standards.

25 ISO/IEC 646:1991, *Information technology - 7-bit coded character set*

26 ISO/IEC 8824-1:1995, *Information technology - Abstract Syntax Notation One (ASN.1) - Part 1: Specification of*
27 *basic notation.*

28 ISO/IEC 8825-2:1995 *Information technology - Specification of ASN.1 encoding rules - Part 2:*

29 *Packed encoding rules (PER)*

30 ISO/IEC 9075:1992, *Information technology - Database languages - SQL.*

31 *ISO/IEC 9597-2, Information technology - Remote database access - Part 2 SQL specialization*

32 ISO/IEC 10646-1:1993, *Information technology - Universal multiple-octet coded character set (UCS) - Part 1:*
33 *Architecture and basic multilingual plane.*

34 ISO/IEC 10646-1:____¹⁾, *Information technology - Universal multiple-octet coded character set (UCS) - Part 1:*
35 *Architecture and basic multilingual plane. Amendment 2. UCS transformation format (UTF-8)*

36 ISO/IEC 13238-1:____¹⁾, *Information technology - Data management export/import facilities - Part 1: Framework.*

37 ¹⁾To be published

1 4 Term(s) and definition(s)

2 For the purposes of this international standard, the following definitions apply. Unless otherwise noted, the
3 definitions are specific to this international standard.

4 4.1 From Other Standards

5 4.1.1 ISO/IEC 9075

6 This part of ISO/IEC 13238 makes use of the following terms defined in ISO/IEC 9075:

7 **collation; collating sequence;**
8 **implementation-defined;**
9 **implementation-dependent;**
10 **null value (null);**
11 **SQL-implementation.**

12 4.1.2 ISO/IEC 8824-1

13 This part of ISO/IEC 13238 makes use of the following terms defined in ISO/IEC 8824-1:

14 **encoding;**
15 **type;**
16 **value.**

17 4.1.3 ISO/IEC 9075

18 This part of ISO/IEC 13238 makes use of the following terms used extensively in ISO/IEC 9075 but not included in
19 the list of formal definitions in that International Standard.

20 Note: ISO/IEC 9075 uses angle brackets to indicate that a term is used in the standard with a special meaning. A term in
21 angle brackets is a BNF non-terminal (node) defined in ISO/IEC 9075:1992; the BNF itself is defined in clause 3.2 of that
22 standard. The same angle bracket convention is used in this part of ISO/IEC 13238.

23 4.1.3.1

24 **SQL-data**

25 Data described by schemas that are under the control of an SQL-implementation in an SQL-environment. (See
26 ISO/IEC 9075 clause 4.14)

27 4.1.3.2

28 **SQL-environment**

29 Comprises an SQL-implementation, zero or more catalogs, zero or more <authorization identifier>s, zero or more
30 <module>s, and the SQL-data described by the schemas in the catalogs. (See ISO/IEC 9075 clause 4.15)

31 4.1.3.3

32 **SQL-schema**

33 A persistent descriptor that includes <schema name> of the SQL-schema, <authorization identifier> of the SQL-
34 schema, <character set name> of the default character set for the SQL-schema, and the descriptor of every
35 component of the SQL-schema. (See ISO/IEC 9075, clause 4.11)

36 Note: the term "SQL schema" is normally referred to in ISO/IEC 9075:1992 as "schema".

37 4.1.3.4

38 **Table**

39 A multi-set of rows. A table is a base table, a viewed table or a derived table. A base table is a persistent table, a
40 global temporary table, a created local temporary table or a declared local temporary table. (See ISO/IEC 9075,
41 clause 4.9)

1 4.1.3.5**2 Column**

3 A multi-set of values that may vary over time. All values of the same column are of the same data type or domain
4 and are values in the same table. (See ISO/IEC 9075, clause 4.8)

5 4.1.3.6**6 SQL-session**

7 A period, which spans the execution of a sequence of consecutive SQL-statements, invoked by a single user from
8 a single SQL-agent or by the direct invocation of SQL.

9 (See ISO/IEC 9075 clause 4.30)

10 4.1.3.7**11 SQL-domain**

12 A set of possible values used to constrain the valid values that can be stored in SQL-data by various operations
13 (see ISO/IEC 9075 clause 4.7)

14 4.1.4 ISO/IEC 15475-1

15 This part of ISO/IEC 13238 makes use of the following terms from ISO/IEC 15475-1 CDIF General Rules for
16 Transfer and Encoding:

17 *N06- Editor's note: They are repeated here verify their applicability*
18 *and to facilitate review of this document* ||

19 4.1.4.1**20 Clear Text**

21 A form of encoding where the resulting physical file is human-readable.

22 4.1.4.2**23 Encoding**

24 An encoding defines how the elements of a syntax are physically represented using an identified character set.
25 Details of representation of the various terminal symbols and data types in the syntax's grammar are provided.

26 4.1.4.3**27 Non-terminal Symbol**

28 A part of the hierarchical definition of a syntax that is further decomposed in the hierarchy.

29 4.1.4.4**30 Production Rule**

31 The definition of a syntactic element in the form of an expression consisting of characters, character strings, and
32 other syntactic elements.

33 4.1.4.5**34 Syntax**

35 A syntax is a definition of the format of information in a transfer. The definition is in the form of a grammar. It is
36 specified with regard to ordering and repetition, down to the level of terminals, but does not specify the
37 representation of any of the terminal objects in the grammar. Those decisions are defined in the encoding of the
38 syntax.

39 4.1.5 ISO/IEC 13238-1

40 This part of ISO/IEC 13238 makes use of the following terms from ISO/IEC 13238-1 Data management
41 Export/Import Standardization Framework:

42 *N07- Editor's note: They are repeated here to facilitate review of*
43 *this document* ||

1 4.1.5.1**2 Transfer file**

3 A file containing data to be interchanged. It is made up of a header, and a number of components. Components
4 contain either data, or data definition data.

5 4.1.5.2**6 DMEI Transfer file**

7 A transfer file conforming to IS 13238

8 4.1.5.3**9 CDIF Transfer file**

10 A transfer file conforming to IS 15475

11 4.1.5.4**12 DMEI SQL transfer file**

13 A file containing data which defines and describes the content of an SQL-database, or a subset of such database.
14 It is made of a header and at least one of two components: transfer file SQL-schema definition and transfer file
15 SQL-data .

16 4.1.5.5**17 DMEI IRDS transfer file**

18 A file containing data which defines and describes the content of an IRD or a subset of an IRD. It is made up of a
19 header, and at least one of multiple components. There are two types of components: transfer file IRD definition
20 component, and transfer file IRD content component.

21 4.1.5.6**22 DMEI Transfer file header**

23 The first part of a transfer file. The header contains data that uniquely defines the transfer file. It also contains
24 details about the source of the transfer file (the exporter) and source-defined parameters.

25 4.1.5.7**26 Export process**

27 The process of generating a transfer file from a source environment.

28 4.1.5.8**29 exporter**

30 The agent of the export process.

31 4.1.5.9**32 import process**

33 The process of incorporating the content of a transfer file into a target environment.

34 4.1.5.10**35 importer**

36 The agent of the import process.

37 4.1.5.11**38 clear text file encoding**

39 A class of techniques for representing data based on first defining a human readable representation using some
40 specific character repertoire and then defining an encoding for that repertoire.

41 4.2 For this standard

42 For the purpose of this part of ISO/IEC 13238 the following definitions apply:

1 4.2.1**2 export/import file**

3 A DMEI SQL transfer file (this definition can be removed if the term "transfer file" replace "export/import file in the
4 whole document.

5 4.2.2**6 transfer file SQL definition**

7 The component of a DMEI SQL transfer file that contains the definition of SQL-data.

8 4.2.3**9 transfer file SQL-data**

10 The component of a DMEI SQL transfer file that contains the SQL-data to be interchanged.

11 4.2.4**12 export SQL-database**

13 The process of generating a DMEI SQL transfer file from a SQL source environment.

14 *N08- Editor's note : This process could be used from another kind*
15 *of environment such as a spreadsheet. This capability has not*
16 *been referenced further in the current version. The full*
17 *implications of this remain to be analyzed.*

18 4.2.5**19 import SQL-database**

20 The process of incorporating the content of a DMEI SQL transfer file into a target SQL environment.

21 *N09- Editor's note: This process could be used into another kind of*
22 *environment such as a spreadsheet. This capability has not*
23 *been referenced further in the current version. The full*
24 *implications of this remain to be analyzed.*

25 4.2.6**26 SQL export/import implementation**

27 A part of a database management system (typically an SQL implementation) that conforms to this part of ISO/IEC
28 13238.

29 4.2.7**30 SQL-definition**

31 A definition of a collection of SQL-data.

32 4.2.8**33 SQL-database**

34 A collection of data comprising the content of all or part of a SQL-definition, and/or a collection of SQL-data.

35 5 Symbols (and abbreviated terms)**36 5.1 Notation****37 5.1.1 ASN.1**

38 A transfer file instance is the encoding of a value of the ASN.1 type SqlTf defined in clause 8, figures 1,2,3 and 4.
39 The way of encoding multiple rows of an SQL table in this part of this International Standard is identical to that used
40 in IS9592-2.

41 Two encodings are specified for such values. These are:

42 a) use of ASN.1 Packed Encoding Rules (See ISO/IEC 8825-2);

- 1 b) use of ASN.1 value notation with subsequent encoding of the resulting characters as specified in this clause ?
 2 of this IS/IEC).

3 These encodings are referred to in this International Standard as <binary packed encoding> and <clear text
 4 encoding> respectively.

5 The file format does not define a representation of the SQL NULL value. Such values are identified in the file by
 6 omission of the corresponding column.

7 **5.1.1 BNF**

8 An extended Bachus Naur Form (BNF) is used to define the structure and describe the sequence of data in the
 9 transfer file. The form of BNF used is that defined in clause 3.2 of ISO/IEC 9075.

10 **5.2 Abbreviations**

11 The following abbreviations are used in this international standard.

12 ASN.1 Abstract Syntax Notation One

13 BMP Basic Multilingual Plane (of ISO/IEC 10646-1)

14 BNF Backus Naur Form

15 EIF Export/import Facility

16 *N10- Made consistent with IRDS part (Facility replaces File)*

17 SQL Database Language SQL

18 UTC Co-ordinated Universal Time

19 UTF Universal Transformation Function

20 **6 Concept and Facilities**

21 The concepts and facilities in this part of ISO/IEC 13238 are related to the transfer file format and the export and
 22 import processes, which are used to export and import SQL-data.

23 **6.1 Transfer File**

24 **6.1.1 Structure**

25 A transfer file contains information, which defines the content of an SQL-database or of a sub-set of an SQL-
 26 database.

27 The transfer file is made up of the following components:

28 a) transfer file header ;

29 b) transfer file SQL-definition;

30 c) transfer file SQL-data .

31 The transfer file header is a mandatory component. Both the export/import SQL-definition and the export/import
 32 SQL-data are optional. However, one of the two must be present.

1 The reason for the optionality of the SQL-definition component is to allow it to be omitted in situations where data
2 conforming to the same SQL-schema definition is being transferred frequently between the same pair of exporter
3 and importer.

4 The reason for the optionality of the SQL-data component is to allow an SQL-definition to be exported without any
5 conforming SQL-data. This approach may be used prior to a set of transfer files being created using the same
6 SQL-definition, but containing different SQL-data.

7 **6.1.2 Transfer file header**

8 The transfer file header contains information about the transfer file, such as date time of file creation, information
9 about the source of the file, such as the person responsible for creation. Further user-defined information may be
10 added.

11 **6.1.3 Transfer file SQL-definition component**

12 The transfer file SQL-definition component consists of the definition of the transfer file SQL-data which may then be
13 exported as other components of a transfer file SQL-data in the same or subsequent transfer files.

14 A transfer file SQL-definition is for a set of SQL-tables that a user of an SQL-implementation has created and that a
15 user wishing to export SQL-data has selected.

16 Each SQL-definition has a name conforming to the SQL rules for names. The exporter of the transfer file in which
17 the SQL-definition is used assigns this name.

18 **6.1.4 Transfer file SQL-data component**

19 This component is the SQL-data. The transfer file SQL-data shall conform (in the SQL sense) to an SQL-definition
20 (if present) which is named in the component header. The SQL-definition may or may not be present in the same
21 transfer file.

22 Note: If an exporter does not want the SQL-data to conform completely to the SQL-definition, then any or all of the rows in the
23 table SQL-Table Constraint (and dependent tables) may be omitted.

24 **6.2 Export/import processes**

25 **6.2.1 Export process**

26 An export process converts a transfer file SQL-definition and the SQL-data selected by an exporter in an SQL-
27 environment into the format defined for the transfer file. The export process also generates the transfer file header
28 component. The export process produces a transfer file containing a transfer file header and either or both of the
29 other two types of components.

30 The physical transfer of a transfer file is not part of the export process.

31 **6.2.2 Import process**

32 An import process converts the content of an identified transfer file into the form required by an SQL-
33 implementation.

34 If the transfer file SQL-definition is included, an import process converts it into the form required by the importing
35 SQL-implementation and enters it into an SQL-catalog in the importing SQL-environment under the name given in
36 the transfer file header component.

- 1 If the transfer file SQL-data component is included and the transfer file SQL-definition component is not included,
 2 an import process makes use of the name for a transfer file SQL-definition given in the SQL-data header
 3 component to locate the export/import SQL-definition in an SQL-catalog in the importing SQL-environment.
- 4 If the transfer file SQL-data component is included, an import process converts it, in accordance with the specified
 5 transfer file SQL-definition, into the form required by the importing SQL-implementation.

6 **7 Specification of the transfer file**

7 **7.1 Codeset**

8 *N11- Editor's Note: This has been made consistent with the IRDS*
 9 *document. However the rest of this document discusses three*
 10 *possible codesets. Is that obsolete and need to be removed,*
 11 *or is it because of some interaction with the SQL standard? In*
 12 *any case Section 7.1 and 8.2.2 need to be made consistent.*

13 **7.1.1 Character Set**

- 14 An SQL transfer file shall consist of a string of the characters from the basic multilingual plane of ISO/IEC 10646-1.
 15 The file shall contain only characters from Table 1 of ISO/IEC 10646-1.

16 **7.1.2 Character Set Encoding**

- 17 Characters in a transfer file shall be encoded using the UCS-2 encoding of ISO/IEC 10646.

18 **7.2 Encoding**

- 19 A transfer file shall contain an encoding of a single value of the ASN.1 type "SqlTf" defined in figures 1,2,3 and 4.
 20 Two different encodings are specified for such values, both of which contain exporter's options. Clause 10 specifies
 21 the conformance requirements for exporters and for importers.
- 22 Note: The effect of these requirements is that there are two conforming classes of export/import implementations (corresponding
 23 to the two different transfer file encodings) that will not interwork, but within any one class all conforming importers will
 24 successfully process all files that a conforming exporter can produce.

25 The two encodings are called:

- 26 a) the export/import-binary-packed encoding
 27 b) the export/import-clear-text encoding

28 These are defined in the following sub-clauses.

29 **7.2.1 Export/import-binary-packed encodings**

- 30 The export/import-binary-packed encoding shall be the BASIC-PER, ALIGNED variant of the ASN.1 Packed
 31 Encoding Rules specified in ISO/IEC 8825-2.

32 Note: This encoding is assigned (by ISO/IEC 8825) the transfer syntax object identifier {joint-iso-itu-t asn1 (1) packed-encoding
 33 (3) basic (0) aligned (0)}.

- 34 The physical form of transfer of the octets generated by this encoding is not specified in this International Standard,
 35 but where the physical transfer mechanism provides a length identification with a granularity of at least ?? octets,

1 only the octets of this encoding shall be placed in the transfer mechanism container. Where the transfer
2 mechanism provides a length indication that is less precise (for example a word count), then the encoding shall be
3 padded with the minimum number of trailing zero octets to satisfy the requirements of the transfer mechanism.

4 *N12- Editor's note Replace ?? above.*

5 7.2.2 Export/import-clear-text encodings

6 The export/import-clear-text encoding is specified in two steps,

- 7 a) the encoding of the value of the ASN.1 type into human-readable text; followed by
- 8 b) the encoding of the human-readable text for transfer.

9 The value of the ASN.1 type "SqlTf" shall be encoded into human-readable text using characters from the Basic
10 Multilingual Plane of ISO/IEC 10646 by representing the value using the ASN.1 value notation specified in ISO/IEC
11 8824-1.

12 Note: All characters in the ASN.1 character set are contained in the Basic Multilingual Plane, and all character strings in the
13 "SqlTf" type are either of type "VisibleString" or of type "BMPString" and hence are present in the Basic Multilingual Plane. Thus
14 this repertoire is sufficient to represent all possible values of "SqlTf" in human-readable form.

15 Where ISO/IEC 8824 specifies that "white-space" or "newline" may appear in the value notation, this shall be
16 interpreted as permission to include zero or more of the characters listed below, in any order.

17 Note: This means that arbitrarily many such characters can appear at the start and end of the export/import-clear-text-encoding,
18 as an exporter's option.

19 Where ISO/IEC 8824 specifies that "white-space" or "newline" shall appear in the value notation, this shall be
20 interpreted as a requirement to include one or more of the characters listed below, in any order.

21 The ISO/IEC 10646 characters to be used (interchangeably) for "white-space" and "newline" are:

22 space (hexadecimal 20);

23 carriage return (hexadecimal 0D)

24 line feed (hexadecimal 0A)

25 ASN.1 comment may be included (as an exporter's option) wherever it is permitted by ISO/IEC 8824-1.

26 The above human-readable encoding shall be encoded into octets in accordance with the specification of UTF-8 in
27 ISO/IEC 10646-1 Amendment 2.

28 Note: This encoding is not self-delimiting, as the number of trailing white-space and newline representations is not determined.

29 This encoding is assigned (by this International Standard) the transfer syntax object identifier {iso standard 13238
30 part (2) oids (1) clear-text (0)}, with object descriptor "SQL transfer file clear-text encoding".

31 The physical form of transfer of the octets generated by this encoding is not specified in this International Standard,
32 but where the physical transfer mechanism provides a length identification with a granularity of at least octets, only
33 the octets of this encoding shall be placed in the transfer mechanism container.

34 Where the transfer mechanism provides a length indication which is less precise (for example a word count), then a
35 minimum number of additional "white-space" characters shall be added to the human-readable encoding before
36 applying UTF-8 in order to satisfy the requirements of the transfer mechanism.

1 8 Transfer file Structure

2 8.1 Syntax for transfer file

3 8.1.1 Content

4 The **transfer file** contains information that defines and describes the content of an SQL-database or a subset of an
5 SQL-database. It is made up of a (1) header component, and at least one of the following transfer components:

6 (2) transfer file SQL-definition and

7 (3) transfer file SQL-data.

8 The **transfer file header** is the first component of a transfer file and it contains information that uniquely defines the
9 transfer file such as the export SQL-environment identification and time stamp. The **transfer file SQL-definition**
10 components are the next optional components of a transfer file and they contain the schema definitions that defines
11 SQL-data (which may or may not be included in the same file). The **transfer file SQL-data components** are the
12 last optional components of a transfer file and they contain the SQL tables that were defined by a previous (in this
13 file or another) transfer file SQL-definition.

14 8.1.2 Definition

```
15 <SqlExportImportFile> ::=
16     <TfHeader><TfComponent> [ <TfComponent> ] ...
```

```
18 <TfComponent> ::=
19     <TfSqlDefinition> | <TfSqlData>
```

21 8.1.3 General Rules

22 1 Both <TfSqlDefinition> and <TfSqlData> are optional. However, at least one of these two shall be included.

23 2 When Present, <TfSqlDefinition> components shall always precede all <TfSqlData> components.

24 3 Any Time or Timestamp data types within the transfer file shall be expressed in UTC.

25 4 <TfSqlHeader> shall contain only characters from Table 1 of ISO/IEC 10646-1.

26 *N13- Editor's Notes: The syntax for the header should be BNF, and*
27 *the encoding CLEAR. To satisfy the design objective of*
28 *standardized headers across transfer file formats, the header*
29 *itself should not be ASN.1, but indicate that the following*
30 *components are ASN.1*

31 8.2 Transfer file header

32 8.2.1 Content

33 The header identifies the file as an SQL transfer file and is mandatory. The header contains the following groups of
34 properties:

35 Transfer file attributes

36 Transfer source attributes

1 [, SOURCE_PROGRAM_VENDOR <ExporterTool Vendor>]
2 <EndSeq>

3
4 <TransferUserParm> : =
5 <TfUserParameters>

6
7 *N14- Editor's notes: Insert here the additional BNF to fully specify*
8 *the header in terms of BNF, if that option is selected.*

9 8.2.3 General Rules

10 1. The constant identifiers shall consist of:

11 <TfTitle>: "ISO/IEC 13238-2:199? SQL TRANSFER FILE" as a constant.

12 *N15- Editor's Note: The previous document had a version number*
13 *for the standard, it has been removed, as the year indicator*
14 *act the same way.*

15 <TfSyntax>: "ASN.1" is the only syntax supported by this part of the International Standard. The field in
16 included to allow for further syntax forms to be prescribed in future versions of this standard or of other
17 standards with a similar file structure.

18 <TfEncoding>: "CLEAR" or "BINARY" are the only tokens encoding supported by this International Standard.
19 The field in included to allow for further token encoding forms to be prescribed in future versions of this
20 standard or of other standards with a similar file structure

21 <TfCodeset> identifies the codeset and the codeset encoding used in the transfer file components.

22 2. The value of the following parameters is set by the export services:

23 Note: The way in which these names are allocated is outside the scope of this standard.

24 <TfTimestamp>: provides the date and time that the file was created in the format of an SQL <timestamp
25 literal> (as defined in ISO/IEC 8601). This is always expressed in UTC

26 <PublisherName>: is used to identify the person or group responsible for the preparation of the transfer file.

27 *N16- Editor's note Clarification of intended use is needed here.*

28 *N17- Editor's note. The context in which <PublisherName> is unique*
29 *needs to be clarified. It is not clear how such names are*
30 *intended to be allocated, although there are various options.*
31 *The above note is the minimum needed.*

32 <PublisherSystemName>: is the unique identifier assigned to the publisher's system, such as an OSI Directory
33 name, a DNS name, or an IP Address.

34 <PublisherVersion>: is a version number assigned to the file by its publisher

35 <ExporterToolPlatform>: identifies the hardware/system software platform of the tool used by the publisher
36 (e.g. Intel Win95, MAC)

37 <ExporterToolName>: is a unique identifier for the exporting tool or product.

38 <ExporterToolVersion> enables the exporter tool name to be qualified with a version identifier.

39 Note: This is chosen by the exporter, and is used to identify the software version of the export facility.

40 <ExporterToolVendor> identifies the provider of the tool used by the publisher

- 1 3. <TfUserParameters>, the user parameters, are optional. The value of the <TfUserParm>, if present, is freely
 2 determined by the exporter. The selection criteria, scope of export and other values may be included as a user
 3 parameter.

4 8.3 Transfer file SQL-definition

5 8.3.1 Content

6 The export/import SQL-definition component either defines the structure of the transfer file SQL-data component if
 7 this is present, or else defines the structure of transfer file SQL-data in another transfer file. In the latter case, the
 8 file serves only to establish a mapping between <SqlDefname> and the SQL-definition for use in other files that
 9 contain only <SQL-data>.

10 The SQL-definition component is made of the following parts:

- 11 1 A component header part
- 12 2 A component content part

13 The component content part contains tables. The syntax for tables is defined in a later section.

14 8.3.2 Definition

```

15 <TfSql Definition> ::=
16     <StartSeq>
17         <TfSql DefinitionHeader>
18             <Sql Definition>
19         <EndSeq>
20 <TfSql DefinitionHeader> ::=
21     COMPONENT_TYPE      <TfComponentType>
22     COMPONENT_TITLE <TfSql DefTitle>
23     COMPONENT_NAME     <Sql DefName>
24 <TfComponentType> ::=
25     DMEI - SQL - DEFINITION
26 <TfSql DefTitle> ::=
27     " ISO/IEC 9075: 1992 SQL DEFINITION"
28 <Sql Definition> ::=
29     <TfTable> [, <TfTable>] ...
30

```

31 8.3.3 General Rules

32 1 The export/import SQL-definition component shall consist of a set of SQL-tables. An exporter must select the
 33 SQL-tables needed from the list in Table 1. Each table has a number, a code and a name, any of which may
 34 be used to identify the table in the transfer file SQL-definition component.

35 Note: Each of these SQL-tables corresponds to a table in the Definition Schema of ISO/IEC 9075 (see clause 2.1.3).

36 2 The exporter shall select the set of transfer file SQL-definition tables that is necessary to provide a complete
 37 and consistent definition of the associated SQL-data to be exported.

1 Notes: (1) For consistency with ISO/IEC 9075, the lexical element underscore <_> is used instead of a space in SQL words
 2 used in Table 1 in the third column for Table names which consist of two or more natural language words. As in the SQL
 3 standard, the hyphen is used when a concept is referred to generically and not as part of the SQL syntax used in this part of
 4 ISO/IEC 13238..

5 Note: (2) The representation of the design of an SQL-database in this part of ISO/IEC 13238 (referred to as an SQL-definition in
 6 this part of ISO/IEC 13238) is based on the representation included in ISO/IEC 9075 for the Definition Schema. This choice of
 7 representation should not be construed as implying that the representation in this part of ISO/IEC 13238 therefore refers
 8 specifically to the Definition Schema. The names of the tables and of the associated columns in each table columns are as
 9 defined for the Definition Schema in ISO/IEC 9078.

No.	Code	Table Name	SQL Definition Schema name
1	CA	SQL_Catalog	None
2	SC	SQL_Schema	SCHEMATA
3	DTD	SQL_Data_Type_Descriptor	DATA_TYPE_DESCRIPTOR
4	DO	SQL_Domain	DOMAINS
5	DC	SQL_Domain_Constraint	DOMAIN_CONSTRAINTS
6	TA	SQL_Table	TABLES
7	VI	SQL_View	VIEWS
8	CO	SQL_Column	COLUMNS
9	TC	SQL_Table_Constraint	TABLE_CONSTRAINTS
10	KCU	SQL_Key_Column_Usage	KEY_COLUMN_USAGE
11	RC	SQL_Referential_Constraint	REFERENTIAL_CONSTRAINTS
12	CC	SQL_Check_Constraint	CHECK_CONSTRAINTS
13	AS	SQL_Assertion	ASSERTIONS
14	CS	SQL_Character_Set	CHARACTER_SETS
15	COL	SQL_Collation	COLLATIONS
16	TP	SQL_Table_Privilege	TABLE_PRIVILEGES
17	CP	SQL_Column_Privilege	COLUMN_PRIVILEGES
18	US	SQL_User	USERS

10

11

Table 1 SQL-definition tables

12

13 3 Column names for each SQL-definition table, as prescribed for the corresponding Definition Schema tables in
 14 ISO/IEC 9075 (see Table 1)

15 4 <SqlDefName>: is an SQL name which identifies the SQL-definition in the transfer file ;

16 Note: This name is chosen arbitrarily by an exporter. The same name should not normally be applied to different values of
 17 <definition> produced by the same exporter. Where an importer is receiving files from several sources, it will in general be
 18 necessary to know the name of the exporter (see <PublisherName> and PublisherSystemName) to make the <SqlDefName>
 19 unambiguous.

20 5 The transfer file SQL-definition is optional. If omitted, then it is assumed that the <SqlDefName> in the header
 21 of the SQL-data component will be recognized by (and known to) the importer, who will provide this
 22 information.

1 8.4 Transfer file SQL-data

2 8.4.1 Content

3 The transfer file SQL-data component consists of SQL-data that has been exported and may be imported.

4 The SQL-data component is made of the following parts:

5 1 A component header part

6 2 A component content part

7 The component content part contains tables. The syntax for tables is defined in a later section.

8 8.4.2 Definition

```

9     <TfSql Data> ::=
10
11         <StartSeq>
12         <TfSql DataHeader>
13         <Sql Data>
14         <EndSeq>
15
16     <TfSql Defi ni ti onHeader> ::=
17         COMPONENT_TYPE          <TfComponentType>
18         COMPONENT_TITLE         <TfSql DataTi tle>
19         COMPONENT_NAME         <Sql Name>
20         COMPONENT_DEFINITION_NAME <Sql DefName>
21
22     <TfComponentType>
23         DMEI - SQL- DATA
24
25     <TfSql DataTi tle> ::=
26         "ISO/IEC 9075:1992 SQL DATA"
27
28     <Sql Data> ::=
29         <TfTable> [, <TfTable>] ...

```

25 8.4.3 General Rules

26 1 <SqlName> : an SQL name which identifies the SQL-data in the transfer file ;

27 *N18- Editor's note <SQL name> is not a term used in 9075.*
 28 *However, it is introduced in 10728. It may not be an*
 29 *appropriate term to use in 13239-2.*

30 Note: This name is chosen arbitrarily by an exporter. The same name should not normally be applied to different values of
 31 <SqlData> produced by the same exporter. Where an importer is receiving files from several sources, it will in general be
 32 necessary to know the name of the exporter (<PublisherName> and/or <PublisherSystemName>) to make the <SqlName>
 33 unambiguous.>

34 2 <SqlDefName>: is an SQL name, which identifies the SQL-definition applicable to this SQL-data component.
 35 Since the transfer file SQL-definition is optional, it is assumed that the <SqlDefName> in the header of the
 36 SQL-data component will be recognized by (and known to) the importer, who will provide this information.

37 3 In the SQL-data, each data value is optional and its name will not be exported when its value is null or zero.
 38 The transfer file will contain relevant information and will not contain large numbers of spaces or zeros.

1 *N19- Editor's note The phrase "or zero" should be deleted from this*
 2 *note. If data values of zero are not included, the importer will*
 3 *not know whether the value is null or zero.*

4 8.4.4 Syntax for transfer file SQL-data

5 8.4.5 Content

6 SQL-definition and SQL-data are transferred using the same tabular structure, referred to as <TfTable>.

7 The Table transfer Structure is made of the following parts:

8 1 a table header part

9 2 a table row part

10 The table row part contains column values for the row.

11 8.4.6 Definition

```

12 <TfTable> ::=
13             <StartSeq>
14             <TfTableHeader>
15             <TfRow> [, <TfRow>] ...
16             <EndSeq>
17 <TfTableHeader> ::=
18     TABLE_NAME    <TfTableName>,
19     TAGS            <TfTableRowTags>
20 <TfTableName> ::=
21     <TableName>
22 <TfTableRowTags> ::=
23     FULL | SHORT | NUMBER | UNLABELLED
24 <TfRow> ::=
25             <StartSeq>
26             <TfColumn> [, <TfColumn>] ...
27             <EndSeq>
28 <TfColumn> ::=
29     [ <TfColumnId> ] <TfColumnValue>
30 <TfColumnId> ::=
31     <TfColumnName> | <TfColumnTag> | <TfColumnNumber>
32

```

33 8.4.7 General Rules

34 *N20- Editor's note It needs to be clarified whether visible string can*
 35 *contain all possible column name characters.*

36 1 The file format does not represent NULL explicitly. This choice does not include the representation of SQL
 37 NULLS as these are not exported. Absence of a column name implies a NULL.

N21- Editor's Note: What happens when columns names are not used

2 <TableName>: The name of the table as supplied by the SQL-environment.

3 <TfColumnName>: The name of the column as supplied by the SQL-environment.

4 <TfColumnValue>: The value of the column as supplied by the SQL-environment

5 <TfColumnTag>: <TBD, see note below>

6 <TfColumnNumber>: <TBD, see note below>

7 <TfTableRowTags>: One of FULL, SHORT, NUMBER OR UNLABELLED constants.

8 FULL means that the actual column names are used.

N22- Editor's notes: For the following option, unambiguous and computable rules must be prepared to specify what the sequence of values is, or it must be specified that the sequence is implementor-defined. This is applicable both in the case of SQL-definition tables, and SQL-data tables

9 UNLABELLED means that the column values are not identified, and are placed in the file in the order of their definition.

N23- Editor's notes: For the following options, unambiguous and computable rules must be prepared to derive the column identifier, This is applicable both in the case of SQL-definition tables, and SQL-data tables

10 SHORT means that either the code in the second column of Table 1 or the first character of the first three words in the column name, are used.

11 NUMBER means that the column position is used as the tag.

12 8 According to the value of <TfTableRowTags>, the following apply

13 **If** <TfTableRowTags> = FULL **then**

14 <TfColumn> ::= <TfColumnId><TfColumnValue>

15 <TfColumnId> ::= <TfColumnName>

16 **EndIf**

17 **If** <TfTableRowTags> = SHORT **then**

18 <TfColumn> ::= <TfColumnId><TfColumnValue>

19 <TfColumnId> ::= <TfColumnTag>

20 **EndIf**

21 **If** <TfTableRowTags> = NUMBER **then**

22 <TfColumn> ::= <TfColumnId><TfColumnValue>

23 <TfColumnId> ::= <TfColumnNumber>

24 **EndIf**

25 **If** <TfTableRowTags> = UNLABELLED **then**

```
1         <TfColumn> ::= <TfColumnValue>
2     EndIf
3
```

4 9 Specification of export/import processes

5 This clause defines a generic SQL PSM interface, which contains the export and the import procedures. The
6 interface is defined as SQL PSM procedures. The conventions used as those defined in international standards
7 ISO/IEC 9075-4.

8
9 These procedures must be invoked in the same way as any other SQL PSM procedures.

10 9.1 Procedures provided

11 Procedures are provided to:

- 12 a) Open the transfer file;
- 13 b) Select the rows to be exported from specified tables;
- 14 c) Export the whole of a database accessible to a user's authorization-id or else export all the tables in a catalog
15 or schema;
- 16 d) Export a set of selected tables and the rows in these table;
- 17 e) Close the transfer file.

18 9.2 Selection of data to be exported

19 Selection takes place within the context of an SQL Connection established by a previous successful call to SQL
20 CONNECT and not terminated by an error condition or a call to SQL DISCONNECT.

21 The SQLEXP procedure exports all the rows in the whole of a database accessible to a user's authorization-id or
22 all the tables in a catalog or in a schema;

23 When not all rows in all tables are to be exported, the required tables are specified by invoking the
24 <SqlSelectForExport> procedure prior to invoking the <SqlExportSelectedData> procedure.

25 Each invocation of a <SqlSelectForExport> procedure selects a set of rows from a named table.

26 The <SqlSelectForExport> procedure may be invoked many times. The effect of multiple calls is to define multiple
27 sets of rows for export by the <SqlExportSelectedData> procedure.

28 Multiple <SqlSelectForExport> procedure calls may name the same table. The effect is to select several sets of
29 rows from the same table. Although a row may thus be selected several times, a selected row is exported exactly
30 once.

31 A previously established set of selection criteria can be cancelled by invoking the <SqlClearExportSelection>
32 procedure.

1 9.3 Open SQL Transfer file

2 9.3.1 Function

3 Invocation of this procedure causes an output file to be opened and the header defined by <TfSqlHeader> to be
4 written to the file.

5 9.3.2 Definition

6 PROCEDURE

7 SqlOpenExportFile

8 (

9	IN: ExpFilePath	String,
10	IN: ExpFileName	String,
11	IN: TfEncoding	String,
12	IN: TfCodeset	String,
13	IN: PublisherVersion	String,
14	IN: TfUserParm	SqlText,
15	OUT: ExpSessionID	SessionID,
16	OUT: RetCode	SqlRetCode

17);

18

19 9.3.3 Input

20 The following parameters must be set as input to the procedure.

21 <ExpFilePath> : when not null, the path to the directory in which the transfer file is to be placed. When null, a
22 default path will be used.

23 <ExpFileName> : when not null, the name to be used for the transfer file . When null, a default file name will be
24 used.

25 <TfEncoding>: the encoding identifier (CLEAR | BINARY) indicating the encoding technique to be used in the
26 transfer file.

27 <TfCodeset>: the codeset and the codeset encoding used in the transfer file components.

28 <PublisherVersion>: is a version number assigned to the file by its publisher

29 <TfUserParm> : when not null, the value is user defined to add extra user information to the transfer file header.

30 9.3.4 Output

31 The following parameters are set by the procedure:

32 <ExpSessionID> : an identifier for the export session established by the <SqlOpenExportFile> call and used to
33 identify the export session on subsequent <SqlSelectForExport>, <SqlClearExportSelection>, SQLExport,
34 <SqlExportSelectedData> and <SqlCloseExportFile> services.

35 <RetCode> : with the following error states:

1	42	User does not have the appropriate privileges
2	nn	The named SQL transfer file is already open
3	nn	Invalid File Name
4	nn	Specified File Path does not exist
5	nn	Other Operating System File Error

6 9.3.5 General Rules

- 7 1 A successful invocation of <Connect> shall be made prior to invoking <SqlOpenExportFile>.
- 8 2 If the file identified by the combination of <ExpFilePath>' and <ExpFileName>' does not exist, it is created. If
9 the file already existed, its contents are replaced by the data exported by the series of SQLEXP and
10 <SqlExportSelectedData> services between a <SqlOpenExportFile> and the following <SqlCloseExportFile>.
- 11 3 The SQL User who invokes this procedure shall have a select privilege on the selected data.
- 12 4 An instance of <TfHeader> is written to the transfer file. Constant fields are given the values specified in
13 clause 8. The variable fields are completed as follows:

14 *N24- Editor's note Some of the fields in the SQL Header are not*
15 *precisely compatible with the IRDS E/I Header. Should we*
16 *make them so ?*

- 17 <TfTimestamp>: to be a time between the start and end of the invocation of the <SqlOpenExportFile>
18 procedure.
- 19 <PublisherName> : the UserID of the SQL User from the SQLOpen procedure invocation that established the
20 current session.
- 21 <PublisherSystemName>: the unique identifier assigned to the publisher's system, such as an OSI Directory
22 name, a DNS name, or an IP Address.
- 23 <ExporterToolName> : an implementor defined name of the tool generating the transfer file.
- 24 <ExporterToolVersion> : an implementor defined version name of the tool generating the transfer file.
- 25 <ExporterToolVendor> : an implementor defined name that identifies the provider of the tool used by the
26 publisher
- 27 <TfUserParameters> the value of the <TfUserParm> parameter supplied to the <SqlOpenExportFile> call.
- 28 6. *An instance of <TfSqlDefinition> is written to the file. ???*

29 9.4 Select For Export Procedure

30 9.4.1 Function

31 This procedure selects for export a set of rows that conform to the selection expression provided from the named
32 table.

33 9.4.2 Format

34 PROCEDURE

```

1      Sql SelectForExport
2      (
3          IN:  ExpSessionID          SessionID,
4          IN:  TableName             Sql Name,
5          IN:  SelectionExpression  String,
6          OUT: RetCode               Sql RetCode
7      );

```

8 9.4.3 Input

9 The following parameters must be set as input to the procedure

10 <ExpSessionID>: the identifier for the export session established by a previous successful invocation of
 11 <SqlOpenExportFile>.

12 <TableName>: the name of a table whose rows are to be selected for export.

13 <SelectionExpression> : an SQL <selection_expression> used by the <SqlSelectForExport> procedure to select
 14 rows that conform to the <selection_expression>.

15 9.4.4 Output

16 The following parameters are set by the procedure:

17 <RetCode> := with the following error states:

18	03	Object selection criteria not satisfied
19	08	Table does not exist;
20	42	User does not have the appropriate privileges
21	07	Invalid column name
22	mm	Invalid selection expression

23 9.4.5 General Rules

- 24 1. <TableName> shall identify an SQL table or view accessible to the user's authorization-id.
- 25 2. <SelectionExpression> shall be a valid <selection_expression> as defined in ISO/IEC 9075-2.
- 26 3. The <SqlSelectForExport> procedure may be invoked many times before the SQLEXP is called.

27 9.5 Clear Export Selection Procedure

28 9.5.1 Function

29 This procedure clears any previously established selections for export.

30 9.5.2 Format

```

31      PROCEDURE
32      Sql ClearExportSelection

```

```

1      (
2          IN:  ExpSessi onID          Sessi onID,
3          OUT: RetCode              Sql RetCode
4      );

```

5 9.5.3 Input

6 The following parameter must be set as input to the procedure

7 <ExpSessionID> : the identifier for the export session established by a previous successful invocation of
8 <SqlOpenExportFile>.

9 9.5.4 Output

10 The following parameters are set by the procedure:

11 <RetCode> : with the following error states:

12 nn No open transfer file.

13 9.5.5 General Rules

14 1. Any current selection is cleared.

15 9.6 Export Procedure

16 9.6.1 Function

17 This procedure exports one of the following:

18 the SQL-data in the tables accessible to the user's authorization-id,

19 the SQL-data in the tables in a named catalog

20 the SQL-data in all tables in a named schema.

21 9.6.2 Definition

```

22  PROCEDURE
23  Sql Export
24  (
25      IN:  ExpSessi onID          Sessi onID,
26      IN:  Catal ogName          Stri ng,
27      IN:  SchemaName           Stri ng,
28      IN:  Sql DefName          Stri ng,
29      IN:  Sql Name             Stri ng,
30      IN:  ExpDefini ti on      Bool ean,
31      IN:  ExpData              Bool ean,
32      IN:  TfTabl eRowTags     Stri ng,
33      OUT: RetCode              Sql RetCode

```

1);

2 **9.6.3 Input..**

3 The following parameters must be set as input to the procedure

4 <ExpSessionID> : the identifier for the export session established by a previous successful invocation of
5 <SqlOpenExportFile>.

6 <CatalogName> : an optional identifier for an SQL-catalog. If <SchemaName> is null, the <CatalogName> may be
7 null.

8 <SchemaName> : an optional identifier of an SQL-schema in the named catalog. The string may be null.

9 <SqlDefName> : the name assigned by the exporter to the transfer file SQL-definition that is being exported.

10 <SqlName> : the name assigned by the exporter to the transfer file SQL-data that is being exported.

11 <ExpDefinition> : when TRUE, then the SQL-definition is to be exported as the second part of the transfer file.
12 When FALSE, then the SQL-definition is not exported and the SQL table and column names in the data are used to
13 identify the structure of the SQL-data to the importer.

14 *N25- Editor's notes: This constraint on TfTableRowTags needs to*
15 *be articulated.*

16 <ExpData> : when TRUE, then the SQL-data as selected or the entire SQL-data are to be exported as the third
17 component of the transfer file.

18 <TfTableRowTags>: One of FULL, SHORT, NUMBER OR UNLABELLED constants.

19 **9.6.4 Output**

20 The following parameter is set by the procedure:

21 <RetCode> : with the following error states:

22 nn No open transfer file.

23 **9.6.5 General Rules**

24 1. If both <CatalogName> and <SchemaName> are both null, then all rows of all tables for which the user
25 has select authorization in all schemas and in all catalogs for which the user has select authorization are
26 written to a transfer file.

27 2. If <CatalogName> is not null and <SchemaName> is null then all rows of all tables for which the user has
28 select authorization in all schemas in the named catalog for which the user has select authorization are
29 written to a transfer file.

30 3. If <CatalogName> is not null and <SchemaName> is not null then all rows of all tables for which the user
31 has select permissions in the named schema in the named catalog for which the user has select
32 authorization are written to a transfer file.

33 4. If the execution of an invocation of the procedure SQLexport is unsuccessful, the value of SQLstate is set
34 to indicate the reason for the failure.

35 5. If the execution of an invocation of the procedure SQLexport is successful, a transfer file is created and the
36 value of SQLstate is set to indicate successful completion.

- 1 6. The exporter shall have a select privilege on the SQL-data to be exported.
- 2 7. Tables and columns may be exported in any sequence, but consistency is advised. As each table is self
3 contained, then the sequence is not significant. Where columns in one table are used to reference similar
4 columns in another table, then it is advisable to define the referenced table first and then to define the
5 referencing table later.
- 6 8. The semantics of the SQL-data contained in a transfer file shall not be dependent on any specific SQL-
7 environment.

8 **9.6.6 Output from export process**

9 As a result of the invocation and successful execution of the procedure SQLexport, a transfer file created and
10 SQLstate is set by the procedure. The alternative values of SQLstate are shown in Table 2.

11

Condition	Class	Sub-condition	Sub-class
data exception	22	invalid parameter value	23
invalid authorization specification	28	(no subclass)	0
invalid catalog name	3D	(no subclass)	0
invalid schema name	3F	(no subclass)	0
no data	2	(no subclass)	0
warning	1	privilege not granted	7
warning	1	query expression too long	00A
invalid encoding type			
successful completion	0	(no subclass)	0

12

13 Note: The entries in this table (except for invalid encoding type) have been extracted from clause 22.1 of ISO/IEC 9075.

14 **Table 2 Possible settings of SQLstate after export process**

15

16 **9.7 SQLExportSelectedData**

17 **9.7.1 Function**

18 This procedure exports rows selected by one or more prior invocations of the <SqlSelectForExport> and not
19 cancelled by a call to <SqlClearExportSelection>.

20 **9.7.2 Format**

```

21 procedure SqlExportSelectedData
22 (
23     IN: ExpSessionID           SessionID,
24     IN: SqlDefName            String,
25     IN: SqlName               String,
26     IN: ExpDefinition         Boolean,

```

```

1         IN:  ExpData          Boolean,
2         OUT: RetCode          RetCode
3     );

```

4 9.7.3 Input

5 The following parameters must be set as input to the procedure

6 <ExpSessionID> : the identifier for the export session established by a previous successful invocation of
7 <SqlOpenExportFile>.

8 <SqlDefName> : the name assigned by the exporter to the transfer file SQL-definition that is being exported.

9 <SqlName> : the name assigned by the exporter to the transfer file SQL-data that is being exported.

10 <ExpDefinition> : when TRUE, then the export definition is to be exported as the second part of the transfer file.
11 When FALSE, then the export definition is not exported and the SQL table and column names in the data are used
12 to identify the structure of the data to the importer.

13 <ExpData> : when TRUE, then the SQL-data as selected or the entire SQL-data are to be exported as the third
14 component of the transfer file.

15 9.7.4 Output

16 The following parameter is set by the procedure:

17 <RetCode> : with the following error states:

18	nn	No successful prior calls to <SqlSelectForExport>.
19	03	No data selected
20	22	Data exception
21	42	User does not have the appropriate privileges.

22 *N26- Editor's note does the complete table for the previous*
23 *procedure apply ?*

25 9.7.5 General Rules

- 26 1. A successful call to <SqlOpenExportFile> and at least one successful call to <SqlSelectForExport> shall be
27 made prior to calling <SqlExportSelectedData>.
- 28 2. Invocation of this procedure generates one or more entries in the transfer file with the structure defined by
29 <TfSqlData >.
- 30 3. The User who invokes this procedure shall have a select privilege on the tables specified by the prior
31 successful calls to <SqlSelectForExport> .

32 9.8 Close SQL Transfer file Procedure

33 9.8.1 Function

34 Invocation of this procedure causes the output file to be closed and the file made accessible by other users.

1 9.8.2 Format

2 Procedure

3 Sql CloseExportFile

```

4 (
5     IN:  ExpSessionID      SessionID,
6     OUT: RetCode          SqlRetCode
7 );

```

8 9.8.3 Input

9 The following parameters must be set as input to the procedure

10 <ExpSessionID> := the identifier for the export session established by a previous successful invocation of
 11 <SqlOpenExportFile>.

12 9.8.4 Output

13 The following parameter is set by the procedure:

14 <RetCode> : with the following error states:

15	25	Invalid Session identifier
16	n1	No successful prior call to <SqlOpenExportFile>'
17	n2	Other Operating System File Error

18 9.8.5 General Rules

19 1. Successful invocations of both <Connect> and <SqlOpenExportFile> shall be made prior to invoking
 20 <SqlCloseExportFile>.

21 2. Following an invocation to <SqlCloseExportFile>, <ExpSessionID> is no longer available for use.

22

23

24

25

26 9.9 Import Procedure

27 9.9.1 Function

28 The import procedure copies data from a transfer file into tables in the SQL-environment of the importer. The data
 29 copied may be any of the following:

30 transfer file SQL-definition;

31 transfer file SQL-data;

32 both a) and b).

1 The SQL-catalog and the SQL-schema in this environment to which the SQL-definition in the transfer file
 2 corresponds are indicated by a parameter. (The schema definition may or may not be specially created for the
 3 SQL-definition in the transfer file.)

4 The tables into which the transfer file SQL-data is copied may be indicated in an SQL-definition component.
 5 Alternatively, the transfer file header shall include a name, which indicates the fully qualified names of the tables. In
 6 this case these tables shall already exist in the SQL-environment.

7 The import process uses the encoding type in the transfer file header component when copying the data in the
 8 remaining two components of the transfer file.

9 9.9.2 Input parameters to import procedure

10 The following enumerated types are defined:

```

11     typedef enum {
12         kCreateMissingTables,
13         kSkipMissingTables,
14         kFailOnMissingTable} MissingTablesController;
15     typedef enum {
16         kUpdateExistingRows,
17         kDontUpdateExistingRows} UpdateExistingRowsController;
18     typedef enum {
19         kSkipFailingRows,
20         kDontSkipFailingRows} SkipFailingRowsController;
21     typedef enum {
22         kCommitAtEndOfFile,
23         kCommitAfterEachTable,
24         kCommitAfterEachTableAndEveryNRows} SqlCommitController;
  
```

25 Parameters are provided to control the following aspects of the import process:

26 whether to create tables that do not exist, to halt the import if a non-existent table is found or merely to skip the
 27 table;

28 whether to update a row that already exists;

29 whether to halt the import if a row insert fails for some other reason or merely to skip the row;

30 the name of a file to which rows in error can be written;

31 whether to commit at the end of the file, or after each table, or after each table and also after a specified number of
 32 rows or after every row.

33 The parameters which have to be set (possibly but not necessarily by a user) prior to the invocation of an import
 34 procedure and are therefore regarded as input to that procedure are the following:

```

35     PROCEDURE
36     SqlImport
37     (
38         IN: CatalogName:           String,
39         IN: SchemaName:           String,
  
```

```

1      IN:  SqlDefName:                String,
2      IN:  Sql Name:                  String,
3      IN:  ImpFileName:                Sql Txt,
4      IN:  ImpFilePath:                Sql Txt,
5      IN:  ExpDefini ti on:            Boolean,
6      IN:  ExpData:                    #      Boolean,
7      IN:  Mi ssi ngTabl esControl Indi cator  Mi ssi ngTabl esControl l er,
8      IN:  UpdateExi sti ngRow sIndi cator    UpdateExi sti ngRow sControl l er,
9      IN:  Ski pFai l i ngRow sIndi cator      Ski pFai l i ngRow sControl l er,
10     IN:  ErrorFi l ePath                Sql Txt,
11     IN:  ErrorFi l eName                Sql Txt,
12     IN:  Commi tControl Indi cator          Sql Commi tControl l er,
13     IN:  Commi tAfter                    long,
14     OUT: TfUserParm:                    String
15     );

```

16 <CatalogName> : an SQL-catalog in the SQL-environment of the importer. This SQL-catalog contains the SQL-
 17 schema that corresponds to the SQL-definition in the transfer file.

18 <SchemaName> : the SQL-schema that contains the SQL-schema that corresponds to the SQL-definition in the
 19 transfer file.

20 <ImpFileName> : a fully specified name for the transfer file may be specified by the importer; when partially
 21 specified, a default path is used (normally relative to the current directory).

22 <ExpDefinition> : not yet defined.

23 <ExpData> : not yet defined.

24 <MissingTablesControllIndicator> : one of kSkipMissingTables, kFailOnMissingTable.

25 <UpdateExistingRowsIndicator >: one of kUpdateExistingRows, kDontUpdateExistingRows.

26 <SkipFailingRowsIndicator > one of kSkipFailingRows,kDontSkipFailingRows.

27 <ErrorFilePath >: the directory into which the error report file is to be placed.

28 <ErrorFileName> : the name of the error report file.

29 9.9.3 Output from import procedure

30 One parameter that has to be set by an import procedure and is therefore regarded as output from that procedure
 31 is the following:

32 SQLSTATE := with the class and subclass values as shown in Table 3

33

Condition	Class	Sub-condition	Sub-class
data exception	22	invalid parameter value	23
invalid authorization specification	28	(no subclass)	0

invalid catalog name	3D.	(no subclass)	0
invalid schema name	3F	(no subclass)	0
no data	2	(no subclass)	0
warning	1	privilege not granted	7
invalid encoding type			
successful completion	0	(no subclass)	0

1

2 Note: The entries in this table (except for invalid encoding type) have been extracted from clause 22.1 of ISO/IEC 9075.

3

N27- Editor's note There is no entry in 9075 for invalid table name (for example) and it needs to be clarified how exceptions such as this are to be handled.

4

5

6

Table 3 Possible settings of SQLSTATE after import procedure

7

N28- Editor's note The following are potential additional error code values:

8

9

mm existing data does not match definition.

10

nn Table in transfer file not in existing database and

11

MissingTablesControllIndicator is kFailOnMissingTable.

12

oo Row failed constraint(s).

13

pp SQL Processor has run out of rollback/recovery resources

14

qq Database full. (Note: this should be an existing SQL stateclass).

15 9.9.4 General Rules

16 The following general rules apply to the Import procedure

17 1. The file identified by the combination of <ImpFilePath> and <ImpFileName> shall exist.

18 2. The importer shall have an appropriate select privilege on the SQL-data being imported.

19

N29- Editor's note Clarification is needed of what is <appropriate>

20 3. An SQL-definition in the transfer file is copied to tables in the SQL-environment of the importer. The names
21 of these tables are qualified by the parameters catalogName and <SchemaName>.

22 4. The import process must use the SQL names are used at an SQL-environment importing the transfer file to
23 identify the definition to the importer.

24 5. SQL-data in the transfer file is copied to tables in the SQL-environment of the importer, identified by the
25 parameters catalogName and <SchemaName>. The tables into which the transfer file SQL-data is copied
26 may be indicated in an SQL-definition component. Alternatively, the transfer file header shall include a
27 name which indicates the fully qualified names of the tables. In this case these tables shall already exist in
28 the SQL-environment

29 6. Case:

30 MissingTablesControllIndicator is kCreateMissingTables: The Import Service shall create all those tables
31 specified in the transfer file SQL Definition section that do not already exist in the database.

- 1 MissingTablesControlIndicator is kSkipMissingTables: No new tables are created. Rows belonging to a table
2 that is not in the target database are copied to the file identified by ErrorFilePath and ErrorFileName in
3 transfer file format.
- 4 MissingTablesControlIndicator is kFailOnMissingTable: No new tables are created. The Import Service fails
5 with error code nn when the first table in the transfer file that is not in the existing database is
6 encountered.
- 7 7. Case:
- 8 UpdateExistingRowsIndicator is kUpdateExistingRows then, when the attempted insertion of a row from the
9 transfer file fails a UNIQUE Constraint, the Import Service shall check that the primary key of the existing
10 table is the same as the primary key of the table being imported. If so, the columns of the existing row with
11 the matching primary key are updated. If not, the row in the transfer file is errored.
- 12 UpdateExistingRowsIndicator is kDontUpdateExistingRows then, when the attempted insertion of a row from
13 the transfer file fails a UNIQUE Constraint, the row is errored.
- 14 8. Case
- 15 SkipFailingRowsIndicator is kSkipFailingRows then the import proceeds even if rows in the transfer file are
16 errored.
- 17 SkipFailingRowsIndicator is kDontSkipFailingRows, the import halts with error code oo when the first row in the
18 transfer file to fail a constraint is encountered. Updates are rolled back to the previous commit point as
19 controlled by CommitControlIndicator (see below).
- 20 9. Case:
- 21 CommitControlIndicator is kCommitAtEndOfFile, the import is committed only after successfully processing the
22 whole file. Note that use of this option with a large transfer file may result in the import failing with error pp
23 because the SQL Processor has run out of rollback/recovery resources.
- 24 CommitControlIndicator is kCommitAfterEachTable, the import is committed only after successfully processing
25 of a complete table.
- 26 CommitControlIndicator is kCommitAfterEachTableAndEveryNRows, the import is committed only after
27 successfully processing of the number of rows given in the CommitAfter parameter. If a row is errored and
28 SkipFailingRowsIndicator is kDontSkipFailingRows, the import is rolled back to the last commit point. Note
29 that if it is desired that every good row up to a fail point should be imported, but none after, the user
30 should specify a value of 1 for the CommitAfter parameter.
- 31 10. Transfer file syntax units in error are copied to the file identified by ErrorFilePath and ErrorFileName in
32 transfer file format.
- 33 11. Errors are reported to the file identified by ErrorFilePath and ErrorFileName as transfer file comments
34 following the syntax unit in error.
- 35 *N30- Editor'snote The following two rules need amending in the*
36 *light of the above new rules.*
- 37 12. The import process may call Commit or Rollback (see ISO/IEC 9075 clause 14) in order to secure the
38 imported data or to nullify the effect of the import, respectively.
- 39 13. The import process shall be considered as a single commit point (see ISO/IEC 9075:1992) that either
40 completes successfully or fails completely.
- 41 14. If the import process fails, then the SQL-data already in the SQL-environment shall be restored to the state
42 prior to the initiation of the import process.

1 15. The sequence of tables in both the SQL-definition and SQL-data components of the file is implementation
2 dependent. The need to sequence information as part of an import process for integrity purposes is a
3 matter for the importing process to determine and the file may be parsed more than once to create the
4 sequence needed by a specific invocation of the import process.

5 16. A row is inserted into the target database for every <row> in the transfer file that meets all the
6 relevant constraints.

7 Note: Between the time at which a transfer file is created by an exporter and the time at which the same file is imported, the
8 table and column definitions may have been modified to reflect new business needs. If these modifications are additive, the
9 SQL-data conforming to an earlier SQL-definition may still be able to be imported and matched by table and column name of the
10 definition followed by the data values.

11

12

1 **10 Notes and Tables**

2 **Table of Illustrations**

3 Error! No table of figures entries found.

4 **Table of Tables**

5 Table 1 SQL-definition tables 15

6 Table 2 Possible settings of SQLstate after export process..... 25

7 Table 3 Possible settings of SQLSTATE after import procedure 30

8

1
2
3
4
5

Annex A (Informative)

Conventions mandated by use of ASN.1

6 *N31- Editor's note This annex must either be made consistent with*
7 *the main body of the standard or else removed.* ||

8 ISO/IEC 8824 ASN.1 provides a standard way to describe data structures, including complex nested structures,
9 using constructs for nesting, iteration and choice. In ASN.1., new types (with a capital initial letter) are defined in
10 terms of other types.

11 The following are mandated by ASN.1:

12 A.1 Built-in typesA.1

13 Built-in types, such as "SEQUENCE" or "CHOICE" are shown in all capital letters, as are other ASN.1 keywords.
14 These types are defined by ASN.1. to provide for nested structures, choice and iteration.

15 A.2 Other primitive types

16 Other primitive types, e.g., "UniversalString" are shown with a leading capital letter. The definitions may also
17 include identifiers and tags. If present, identifiers appear to the left of a type name (e.g., "header",
18 "descriptorName") and begin with a lower case letter with optional upper case capitals on following words.

19 A.3 Explanatory comments

20 A transfer file using clear text encoding may contain comments as a double hyphen followed by any text to the end
21 of the line. These comments are not normative and have no effect on the encoding.

22 A.4 Other formatting characters

23 Subsequent spaces after a space are ignored. Line feed and/or carriage return codes are considered as a space.
24 Indentation has no relevance and has been used for human readability. When indentation is used for human
25 readability, then the encoding style is referred to in this part of ISO/IEC 13238 as "clear text".

26 A.5 Example of ASN.1 syntax

27 An example of the use of the part of ASN.1 that is used by this part of ISO/IEC 13238 is given in ISO/IEC 8824-1.
28 The example is based on personnel data.

1
2 **Annex B**
3 **(normative)**

4
5 **ASN.1 Definition of the Transfer Format**

6 *N32- Editor's note This annex is currently:*

- 7 a) *Not harmonized with the rest of the document;*
8 b) *Not harmonized with the use of ASN.1 in the recent RDA*
9 *document;*
10 c) *Not harmonized with the recent improvements to ASN.1*
11 *for "tables".*

12 **Syntax for transfer file**

13 =====
14 001 SqlTf{iso standard 13238 part (2) modules (0) main (0)}
15 002 DEFINITIONS AUTOMATIC TAGS
16 003 ::=
17 004 BEGIN
18 005 SqlTf ::= SEQUENCE
19 006 {header TfSqlHeader,
20 006 CHOICE
21 007 [0] definition TfSqlDefinition ,
22 008 [1] SqlData TfSqlData
23 009 [2] definition TfSqlDefinition SqlData TfSqlData
24 010 }
25 =====

26 **Transfer file header**

27 =====
28 010 TfSqlHeader ::= SEQUENCE
29 011 {title VisibleString ("ISO SQL Transfer file Version 1"),
30 012 syntaxIdentifier VisibleString ("ASN.1"),
31 013 columnEncodingIdentifier TfEncoding, -- see below
32 014 valueCharacterSet TfCharSet, -- see below
33 015 systemParm TfSystem, -- see below
34 016 userParm VisibleString OPTIONAL
35 017 }
36 018 TfEncoding ::= VisibleString
37 019 ("Full" | -- full column name
38 020 "Short" | -- short code

```

1      021 "Number" |          -- column numbers
2      022 "Binary")         -- full ISO/IEC 8825 binary packed encoding
3      023
4      024      TfCharSet ::= VisibleString
5      025      ("ASCII" | -- ISO/IEC 646
6      026      "BMP" |      -- ISO/IEC 10646 Basic Multilingual Plane
7      027      "Universal")      -- ISO 8824 universal string
8      028
9      029 TfSystem ::= SEQUENCE
10     030      {Sql defname VisibleString,
11     031      Sqlname VisibleString OPTIONAL,
12     032 timeStamp GeneralizedTime, -- date & time
13     033      exporterName VisibleString OPTIONAL
14     034      exporterVersion VisibleString OPTIONAL
15     035      publisherName VisibleString OPTIONAL
16     036 }
17     =====

```

18 Syntax for transfer file SQL-definition

```

19     =====
20     037 TfSqlDefinition ::= SEQUENCE          -- component 2
21     038 {title VisibleString ("ISO SQL-DEFINITION"),
22     039 SqlDefinition SEQUENCE OF (1..MAX) Table
23     040 }
24     =====

```

25 Syntax for transfer file SQL-data

```

26     =====
27     060 TfSqlContent ::= SEQUENCE
28     061      {title VisibleString ("ISO SQL-data"), -- component 3
29     062      Sqldata SEQUENCE OF (1..MAX) Table      -- reuse Table
30     063 }
31     064 END
32     =====

```

33 Syntax for transfer file Transfer Table

```

34     =====
35     041 Table ::= SEQUENCE
36     042 {tableName VisibleString,

```

```

1      043 rows
2      044 }
3      045 R-ExecutedDBL-Result ::= SEQUENCE
4      046 {      SqlDBLResultSpecification [1] SqlDBLResultSpecification OPTIONAL,
5      047      listOfResultValues [2] SEQUENCE OF ResultValues OPTIONAL
6      048 }
7      049
8      050 RDCharString ::= SEQUENCE
9      051 {      text [0] OCTET STRING,
10     052      charset [1] OBJECT IDENTIFIER OPTIONAL
11     053 }
12     054 SqlDataTypeDescriptor ::= SEQUENCE
13     055 {      nullable [0] BOOLEAN DEFAULT TRUE,
14     056      colName CHOICE
15     057 {      vstring [1] VisibleString,
16     058      rstring [2] RDCharString
17     059 }      OPTIONAL,
18     060      typeDescriptor CHOICE
19     061      {      characterType [5] SEQUENCE
20     062      -- Sql Type: character
21     063      {      charSet OBJECT IDENTIFIER OPTIONAL,
22     064      length INTEGER,
23     065      fixedLengthEncoding BOOLEAN,
24     066      characterSetCatalog [0] RDCharString OPTIONAL,
25     067      characterSetSchema [1] RDCharString OPTIONAL,
26     068      characterSetName [2] RDCharString OPTIONAL,
27     069      collationCatalog [3] RDCharString OPTIONAL,
28     070      collationSchema [4] RDCharString OPTIONAL,
29     071      collationName [5] RDCharString OPTIONAL
30     072      },
31     073      numericType [6] SEQUENCE
32     074      -- Sql Type: numeric
33     075      {      precision INTEGER,
34     076      scaleINTEGER
35     077      },
36     078      decimalType [7] SEQUENCE
37     079      -- Sql Type: decimal
38     080      {      precision INTEGER,
39     081      scaleINTEGER
40     082      },
41     083      integerType [8] SEQUENCE
42     084      -- Sql Type: integer

```

```

1      085      {    precisi on INTEGER,
2      086          precisi onBase  ENUMERATED
3      087          {    bi nary    (0),
4      088              deci mal    (1)
5      089          }
6      090      },
7      091      smallIntType  [9] SEQUENCE
8      092      -- Sql Type: smallInt
9      093      {    precisi on INTEGER,
10     094          precisi onBase  ENUMERATED
11     095          {    bi nary    (0),
12     096              deci mal    (1)
13     097          }
14     098      },
15     099      floatType  [10] SEQUENCE
16     100      -- Sql Type: float
17     101      {    manti ssaPreci si on  INTEGER,
18     102          maxExponent  INTEGER
19     103      },
20     104      realType  [11] SEQUENCE
21     105      -- Sql Type: real
22     106      {    manti ssaPreci si on  INTEGER,
23     107          maxExponent  INTEGER
24     108      },
25     109      doublePreci si onType  [12] SEQUENCE
26     110      -- Sql Type: doublePreci si on
27     111      {    manti ssaPreci si on  INTEGER,
28     112          maxExponent  INTEGER
29     113      },
30     114      dateTi meType  [13] SEQUENCE
31     115      -- Sql type: datetime
32     116      {    dateTi meQual i fier  ENUMERATED
33     117          {    date (1),
34     118              ti me (2),
35     119              ti meStamp (3),
36     120              ti meWi thTi meZone (4),
37     121              ti meStampWi thTi meZone (5)
38     122          },
39     123          fracti onal SecondsPreci si onINTEGER OPTIONAL
40     124      },
41     125      intervalType  [14] SEQUENCE
42     126      -- Sql Type: interval

```

```

1      127      { intervalQualifier ENUMERATED
2      128          { year (1),
3      129              month(2),
4      130              day (3),
5      131              hour (4),
6      132              minute (5),
7      133              second (6),
8      134              yearToMonth (7),
9      135              dayToHour (8),
10     136              dayToMinute (9),
11     137              dayToSecond (10),
12     138              hourToMinute (11),
13     139              hourToSecond (12),
14     140              minuteToSecond (13)
15     141          },
16     142          leadingFieldPrecision INTEGER,
17     143          fractionalSecondsPrecision INTEGER OPTIONAL
18     144      },
19     145      varcharType [15] SEQUENCE
20     146      -- Sql Type: varchar
21     147      { charSet OBJECT IDENTIFIER OPTIONAL,
22     148          length INTEGER,
23     149          characterSetCatalog [0] RDCharString OPTIONAL,
24     150          characterSetSchema [1] RDCharString OPTIONAL,
25     151          characterSetName [2] RDCharString OPTIONAL,
26     152          collationCatalog [3] RDCharString OPTIONAL,
27     153          collationSchema [4] RDCharString OPTIONAL,
28     154          collationName [5] RDCharString OPTIONAL
29     155      },
30     156      bitType [16] SEQUENCE
31     157      -- Sql Type: bit
32     158      { length INTEGER
33     159      },
34     160      bitVaryingType [17] SEQUENCE
35     161      -- Sql Type: bit varying
36     162      { length INTEGER
37     163      }
38     164      }
39     165  }
40     166
41     167  SqlDBLException ::= SEQUENCE
42     168  { SqlSTATE CHOICE

```

```

1   169     {   vstring   [0] VisibleString,
2   170         rstring   [5] RDCharString
3   171     }
4   172     Sql CODE    [1] INTEGER OPTIONAL,
5   173     Sql ErrorText CHOICE
6   174     {   vstring   [2] VisibleString,
7   175         rstring   [4] RDCharString
8   176     }
9   177     Sql Diagnostics [3] Sql Diagnostics OPTIONAL
10  178 }
11  179 Sql Diagnostics ::= SEQUENCE
12  180 {   rowCount    [0] INTEGER OPTIONAL,
13  181     commandFunction [1] RDCharString OPTIONAL,
14  182     dynamicFunction [2] RDCharString OPTIONAL,
15  183     exceptionList  [3] SEQUENCE OF ExceptionInfo
16  184 }
17  185 ExceptionInfo ::= SEQUENCE
18  186 {   returnedSql STATE    [0] RDCharString,
19  187     classOrigin   [1] RDCharString,
20  188     subclassOrigin [2] RDCharString,
21  189     messageText   [3] RDCharString OPTIONAL,
22  190     serverName    [4] RDCharString OPTIONAL,
23  191     constraintCatalog [5] RDCharString OPTIONAL,
24  192     constraintSchema [6] RDCharString OPTIONAL,
25  193     constraintName  [7] RDCharString OPTIONAL,
26  194     catalogName    [8] RDCharString OPTIONAL,
27  195     schemaName     [9] RDCharString OPTIONAL,
28  196     tableName      [10] RDCharString OPTIONAL,
29  197     columnName     [11] RDCharString OPTIONAL,
30  198     cursorName     [12] RDCharString OPTIONAL
31  199 }
32  200 Sql DBLResultSpecification ::= SEQUENCE
33  201 {   listOfSqlDataTypeDescriptor [0] SEQUENCE OF SqlDataTypeDescriptor
34  202 }
35  203 Sql DBLResultValues ::= Sql ValueList
36  204
37  205 Sql Value ::= SEQUENCE
38  206 {   dataItem CHOICE
39  207     {   characterItem [0] OCTET STRING,
40  208         numericItem  [1] INTEGER,
41  209         decimalItem  [2] INTEGER,
42  210         integerItem  [3] INTEGER,

```

```
1      211          smallIntItem [4] INTEGER,
2      212          floatItem [5] REAL,
3      213          realItem [6] REAL,
4      214          doublePrecisionItem [7] REAL,
5      215          dateTimeItem [8] VisibleString,
6      216          intervalItem [9] VisibleString,
7      217          varcharItem [10] OCTET STRING,
8      218          bitItem [11] BIT STRING,
9      219          bitVarItem [12] BIT STRING
10     220      }          OPTIONAL,
11     221          indicator [30] INTEGER OPTIONAL
12     222 }
13     223 SqlValueList ::= CHOICE
14     224 { listOfSqlValue [1] SEQUENCE OF SqlValue
15     225 }
16     226
```

Annex C (informative)

Sample Transfer file Format

5 C.1 Purpose of this Annex

6 This informative annex is provided to illustrate the mechanism of using this part of ISO/IEC 13238. The steps
7 needed to create a transfer file are documented for information purposes only. Many of these steps may be
8 undertaken before a user needs to create a transfer file. All are listed to show the context in which a transfer file will
9 be used. The steps are as follows.

10 C.1.1 Identify requirement

11 Design a database to support a business requirement using a specific implementation design. A database may be
12 based on an International Standard or it may be an ad-hoc requirement as in the case of this example (see C.2 for
13 the sample business requirement and implementation design used in the example).

14 C.1.2 Create schema

15 Record the application schema as rows in the SQL-definition tables (see C.3 for the sample SQL-definition table
16 content).

17 C.1.3 Populate SQL-definition tables with requirement

18 Record the sample application data structure in the SQL-definition (see C.4 for the sample application table
19 values).

20 C.1.4 Export SQL-definition

21 Export the SQL-definition for the application using an export process.. The actual definition of the relevant data is
22 illustrated in C.5.

23 C.1.5 Export SQL-data

24 Export the SQL-data using an export process. The file content for the relevant data is illustrated in C.6.

25 C.2 Description of sample application

26 C.2.1 Business requirement

27 The business requirement is to be able to record and manipulate data regarding suppliers, purchase orders and
28 products. The types of data (referred to as "entity types" or "business object classes" have been identified and
29 named as:-

30 E1 Supplier

1 E2 Purchase order

2 E3 Product

3 These entity types have been further defined with the identification of the natural relationships between the entity
4 types as:

5 R1 Supplier may have many Products

6 R2 Purchase Order may cover many Products

7 R3 Product may be contained in many Purchase Order

8 Each entity type has been defined in terms of its attributes that have been populated as the following requirement:

9 E1 Product

10 A1 Product_Code

11 A2 Product_Description

12 E2 Supplier

13 A3 Supplier_Code

14 A4 Supplier_Name

15 A5 Product_count

16 E3 Purchase_order

17 A6 Purchase_order_no

18 A7 Supplier_code

19 A8 Product_code

20 This requirement may be represented using a conventional entity relationship diagram..

21 **C.2.2 Implementation requirement**

22 This business requirement is designed as for implementation with the many to many relationship between Product
23 and Purchase order resolved with an entity type called Purchase Order Item.

24 The database structure has the requirement to support the following data:_

25 E1 Product

26 E2 Supplier

27 E3 Purchase_Order

28 E4 Purchase_Order_Item

29 R1 Supplier may have many Products

30 R2 Purchase Orders may be sent to many Suppliers

- 1 R3 Product may be listed on many Purchase Orders
- 2 R4 Purchase Order may contain many Purchase Order Items
- 3 The structure required to implement this representation is as follows.
- 4 E1 Product
 - 5 A1 Product_code
 - 6 A2 Product_description
- 7 E2 Supplier
 - 8 A3 Supplier_code
 - 9 A4 Supplier_name
- 10 A5 Product_count
- 11 E3 Purchase_Order
 - 12 A6 Purchase_order_no
 - 13 A7 Supplier_code
 - 14 A8 Date_of_order
- 15 E4 Purchase_Order_Item
 - 16 A9 Purchase_order_no
 - 17 A10 Item_no
 - 18 A11 Product_Code
 - 19 A12 No_ordered

20 **C.3 Representation of the design of an SQL-database**

21 An SQL-database is designed by the specification of SQL tables. An SQL-definition also consists of a set of SQL-
22 tables, but these tables do not relate to a specific application but to the set of SQL-objects which are necessary to
23 specify any set of SQL-data.

24 It should be noted that ISO/IEC 9075 distinguishes between the Definition Schema and an Information Schema.
25 The standards states

26 "An Information Schema consists of views (namely a set of viewed tables) .The only purpose of the Definition
27 Schema is to provide a data model to support the Information Schema. An SQL-implementation need do no more
28 than simulate the existence of the Definition schema as viewed through the Information Schema views."

29 The following representation illustrates each SQL-definition table considered necessary to support the SQL transfer
30 file with the data necessary to represent the design of the simple example introduced in C.2.

31 The following conventions should be noted:

- 1 a) ISO/IEC 9075 allows many of the constructs (such as table, column, constraint) to be in different schema and
 2 also in different catalogs. For simplicity, the columns which are required to support this functionality are
 3 omitted from this example. This implies that all SQL-objects in this example are in the same schema and in the
 4 same catalog.
- 5 b) Names of the tables in the SQL-definition are expressed in a capitalized form of the name shown in figure 4.
 6 The singular form is used rather than the plural form in ISO/IEC 9075. Names of the columns in the SQL-
 7 definition are also expressed in capitals.
- 8 c) Words which are part of the SQL standard are fully capitalized in this example.
- 9 d) Names of constructs relevant to this example are written with initial capital letters only.
- 10 e) Not all tables in figure 4 are illustrated in this example.

11 C.3.1 SQL_DOMAIN

DOMAIN_NAME	DOMAIN_DEFAULT
Product_code	
Supplier_code	
Purchase_order_no	

12 C.3.2 SQL_DOMAIN_CONSTRAINT

13 Not illustrated in this example.

14 C.3.3 SQL_TABLE

TABLE_NAME	TABLE_TYPE
Product	BASE TABLE
Supplier	BASE TABLE
Purchase Order	BASE TABLE
Purchase Order item	BASE TABLE

15 C.3.4 SQL_VIEW

16 Not illustrated in this example

17 C.3.5 SQL_COLUMN

TABLE_NAME	COLUMN_NAME	ORDINAL_POSITI ON	DOMAIN_NAME
Product	Product_code	1	Product_code
Product	Product_description	2	
Supplier	Supplier_code	1	Supplier_code
Supplier	Supplier_description		
Purchase_order	Purchase_order_no	1	Purchase_order_no
Purchase_order	Supplier_code	2	Supplier_code
Purchase_order	Date_of_order	3	
Purchase_order_item	Purchase_order_no	1	Purchase_order_no

Purchase_order_item	Item_no	2	
Purchase_order_item	Product_code	3	Product_code
Purchase_order_item	No_ordered	4	

1 C.3.6 SQL_VIEW_TABLE_USAGE

2 Not illustrated in this example

3 C.3.7 SQL_VIEW_COLUMN_USAGE

4 Not illustrated in this example

5 C.3.8 SQL_TABLE_CONSTRAINT

CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	IS_DEFERRABLE	INITIALLY_DEFERRED
PR1	PRIMARY KEY	Product	NO	NO
SU1	PRIMARY KEY	Supplier	NO	NO
PO1	PRIMARY KEY	Purchase_order	NO	NO
PO2	FOREIGN_KEY	Purchase_order	YES	NO
POI1	PRIMARY KEY	Purchase_order_item	NO	NO
POI2	FOREIGN KEY	Purchase_order_item	YES	NO
POI3	FOREIGN KEY	Purchase_order_item	YES	NO

6 C.3.9 SQL_KEY_COLUMN_USAGE

CONSTRAINT_NAME	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
PR1	Product	Product_code	1
SU1	Supplier	Supplier_code	1
PO1	Purchase_order	Purchase_order_no	1
PO2	Purchase_order	Supplier_code	1
POI1	Purchase_order_item	Purchase_order_no	1
POI1	Purchase_order_item	Item_no	2
POI2	Purchase_order_item	Purchase_order_no	1
POI3	Purchase_order_item	Product_code	1

7 C.3.10 SQL_REFERENTIAL_CONSTRAINT

CONSTRAINT_NAME	MATCH_OPTION	UPDATE_RULE	DELETE_RULE
PO2	FULL	SET DEFAULT	SET DEFAULT
POI2	FULL	SET DEFAULT	CASCADE
POI3	FULL	SET DEFAULT	SET DEFAULT

8 C.3.11 SQL_CHECK_CONSTRAINT

9 Not illustrated in this example

1 **C.3.12 SQL_CHECK_TABLE_USAGE**

2 Not illustrated in this example

3 **C.3.13 SQL_CHECK_COLUMN_USAGE**

4 Not illustrated in this example

5 **C.3.14 SQL_ASSERTION**

6 Not illustrated in this example

7 **C.3.15 SQL_CHARACTER_SET**

CHARACTER_SET_NAME	FORM_OF_USE	NO_OF_CHARACTERS
Norwegian Riksmål	Standard	29

8 **C.3.16 SQL_COLLATION**

9 Not illustrated in this example

10 **C.3.17 SQL_USER**

USER_NAME
Z6AWPQ
YTWEN5
øWVA5P

11 **C.3.18 SQL_TABLE_PRIVILEGE**

GRANTOR	GRANTEE	TABLE_NAME	PRIVILEGE_TY PE	IS_GRANTABLE
Z6AWPQ	YTWEN5	Product	SELECT	NO
Z6AWPQ	øWVA5P	Supplier	SELECT	NO
Z6AWPQ	YTWEN5	Purchase_Order	SELECT	NO
Z6AWPQ	øWVA5P	Purchase_Order_Item	SELECT	NO

12 **C.3.19 SQL_COLUMN_PRIVILEGE**

13 Not illustrated in this example

14 **C.4 Specification of sample SQL-data**

15 The definition in C.3 is a representation of the SQL-definition for a simple four table database. These four tables
 16 are here identified as "application tables" to distinguish them from the SQL-definition tables listed in C.3.

17 The application tables and other associated data, as defined in C.3 as the content of selected SQL

18 definition tables are illustrated here with sample content for completeness.

1 **C.4.1 Product**

Product_code	Product_description
FX198A	Ostehpvel
FV236J	Brpdkniv
FE437P	Tallerken
TN328D	BlΔbør

2 **C.4.2 Supplier**

Supplier_code	Supplier_description
169235	Olsen
174763	Jensen
549362	Hansen
589267	Petersen

3 **C.4.3 Purchase_order**

Purchase_order_no	Supplier_code	Date_of_order
658	169235	23.12.1999
659	169235	23.12.1999

4 **C.4.4 Purchase_order_item**

Purchase_order_nu mber	Item_no	Product_code	No_ordered
658	1	TN328D	73
658	2	FV236J	29
659	1	TN328D	171
659	2	FE437P	143

5 **C.5 Specification of a transfer file in ASN.1**

N33- Editor's note The actual ASN.1 value notation needs aligning with the corrections to the type notation in the body of the standard.

9 **C.5.1 Overview**

10 This SQL-definition has been reduced in size by using a more compact representation.

11 **C.5.2 Header**

12 The first component of a transfer file conforms to figure 2 in the main body of this part of ISO/IEC 13238.

```

13 =====
14 {header
15 {title "ISO SQL Transfer file", --begin header
16 syntax identifier "ASN.1",
17 encoding identifier "Clear",

```

```

1  valueCharacterSet    "BMP",
2  systemParm          --sequence as follows
3  {SQLdefname         "9075 SQL-definition"
4    SQLname           "9075 SQL-data"
5    timeStamp         "19991225091215"
6    exporterName     "Smith"
7    exporterVersion   "2.3"
8    publisherName     "EI_Systems AB"
9  } -- end system parameters
10 userParm             "Sample user parameter"},
11 =====

```

12 C.5.3 SQL-definition expressed in ASN.1

13 This part consists of each SQL table definition needed to define the tables in C.3 and their content. Only tables with
 14 content in C.3 are included here. Only non-null columns are included here.

15 The explanatory notes at the right hand end of each line are shown in full for the first table, in part for the second
 16 table, in a further reduced form for the third table, and omitted for the third and subsequent tables.

```

17 =====
18 {definition          --part 2 of 3
19 {title {"ISO SQLdefintion"}
20 Sql definition
21 { --begin table set
22 {tablename "SQL_DOMAIN"          --begin table in C.3.1
23 { --begin row 1
24 {tag {f"DOMAIN_NAME"},          value{us "Product_code"}},      -- col 1
25 }, --end column last
26 }, --end row 1, not last
27 { --begin row 2
28 {tag {f"DOMAIN_NAME"},          value{us "Supplier_code"}},      -- col 1
29 }, --end column last
30 }, --end row 2, not last
31 { --begin row 1
32 {tag {f"DOMAIN_NAME"},          value{us "Purchase_order_no"}},  -- col 1
33 }, --end column last
34 }, --end row 3, last
35 {tablename "SQL_TABLE"          --begin table in C.3.3
36 { --begin row 1
37 {tag {f"TABLE_NAME"},            value{us "Product"}},          -- col 1
38 {tag {f"TABLE_TYPE"},            value{us"BASE TABLE"}        -- col 2
39 }, --end column last
40 }, --end row 1, not last
41 { --begin row 2

```

```

1      {tag {f"TABLE_NAME"},                value{us "Supplier"}},          col 1
2      {tag {f"TABLE_TYPE"},                value{us "BASE TABLE"}      -- col 2
3      }, --end column last
4      }, --end row 2, not last
5      {
6          --begin row 3
7      {tag {f"TABLE_NAME"},                value{us "Purchase Order"}},  -- col 1
8      {tag {f"TABLE_TYPE"},                value{us "BASE TABLE"}      -- col 2
9      }, --end column last
10     }, --end row 3, not last
11     {
12         --begin row 4
13     {tag {f"TABLE_NAME"},                value{us "Purchase_Order_item"}}, -- col 1
14     {tag {f"TABLE_TYPE"},                value{us "BASE TABLE"}      -- col 2
15     }, --end column last
16     }, --end row 4, last
17     {tablename "SQL_COLUMN"              --begin table in C.3.5
18     {
19         {tag {f"TABLE_NAME"},            value{us "Product"} },
20         {tag {f"COLUMN_NAME"}, value{us"Product_Code"} },
21         {tag {f"ORDINAL_POSITION"}, value{us "1"} },
22         {tag {f"DOMAIN_NAME"}, value{us"Product_Code"} },
23     }, --end row 1, not last
24     {
25         {tag {f"TABLE_NAME"},            value{us "Product"} },
26         {tag {f"COLUMN_NAME"}, value{us"Product_Description"} },
27         {tag {f"ORDINAL_POSITION"}, value{us "2"} },
28     }, --end row 2, not last
29     {
30         {tag {f"TABLE_NAME"},            value{us "Supplier"} },
31         {tag {f"COLUMN_NAME"}, value{us"Supplier_Code"} },
32         {tag {f"ORDINAL_POSITION"}, value{us "1"} },
33         {tag {f"DOMAIN_NAME"}, value{us"Supplier_Code"} },
34     }, --end row 3, not last
35     {
36         {tag {f"TABLE_NAME"},            value{us "Supplier"} },
37         {tag {f"COLUMN_NAME"}, value{us"Supplier_Description"} },
38         {tag {f"ORDINAL_POSITION"}, value{us "2"} },
39     }, --end row 4, not last
40     {
41         {tag {f"TABLE_NAME"},            value{us "Purchase_Order"} },
42         {tag {f"COLUMN_NAME"}, value{us"Purchase_Order_No"} },
43         {tag {f"ORDINAL_POSITION"}, value{us "1"} },
44         {tag {f"DOMAIN_NAME"}, value{us"Purchase_Order_No"} },
45     }, --end row 5, not last
46     {
47         {tag {f"TABLE_NAME"},            value{us "Purchase_Order"} },
48         {tag {f"COLUMN_NAME"}, value{us"Supplier_Code"} },
49         {tag {f"ORDINAL_POSITION"}, value{us "2"} },
50         {tag {f"DOMAIN_NAME"}, value{us"Supplier_Code"} },

```

```

1      }, --end row 6, not last
2      {   {tag {f"TABLE_NAME"},          value{us "Purchase_Order"} },
3      {tag {f"COLUMN_NAME"}, value{us "Date_of_Order"} },
4      {tag {f"ORDINAL_POSITION"}, value{us "3"} },
5      }, --end row 7, not last
6      {   {tag {f"TABLE_NAME"},          value{us "Purchase_Order_Item"} },
7      {tag {f"COLUMN_NAME"}, value{us "Purchase_order_no"} },
8      {tag {f"ORDINAL_POSITION"}, value{us "1"} },
9      {tag {f"DOMAIN_NAME"}, value{us "Purchase_order_no"} },
10     }, --end row 8, not last
11     {   {tag {f"TABLE_NAME"},          value{us "Purchase_Order_Item"} },
12     {tag {f"COLUMN_NAME"}, value{us "Item_no"} },
13     {tag {f"ORDINAL_POSITION"}, value{us "2"} },
14     {tag {f"DOMAIN_NAME"}, value{us "NULL"} },
15     }, --end row 9, not last
16     {   {tag {f"TABLE_NAME"},          value{us "Purchase_Order_Item"} },
17     {tag {f"COLUMN_NAME"}, value{us "Product_code"} },
18     {tag {f"ORDINAL_POSITION"}, value{us "3"} },
19     {tag {f"DOMAIN_NAME"}, value{us "Product_code"} },
20     }, --end row 10, not last
21     {   {tag {f"TABLE_NAME"},          value{us "Purchase_Order_Item"} },
22     {tag {f"COLUMN_NAME"}, value{us "No_ordered"} },
23     {tag {f"ORDINAL_POSITION"}, value{us "4"} },
24     {tag {f"DOMAIN_NAME"}, value{us "NULL"} },
25     }, --end row 11, last
26     }, --end table
27     {tablename "SQL_TABLE_CONSTRAINT"          --begin table in C.3.8
28     {   {tag {f"CONSTRAINT_NAME"},          value{us "PR1"}},
29     {tag {f"CONSTRAINT_TYPE"},          value{us "PRIMARY KEY"} },
30     {tag {f"TABLE_NAME"},          value{us "Product"} },
31     {tag {f"IS_DEFERRABLE"},          value{us "NO"} },
32     {tag {f"INITIALLY_DEFERRED"},          value{us "NO"} },
33     },
34     {   {tag {f"CONSTRAINT_NAME"},          value{us "SU1"} },
35     {tag {f"CONSTRAINT_TYPE"},          value{us "PRIMARY KEY"} },
36     {tag {f"TABLE_NAME"},          value{us "Supplier"} },
37     {tag {f"IS_DEFERRABLE"},          value{us "NO"} },
38     },
39     },
40     {   {tag {f"CONSTRAINT_NAME"},          value{us "P01"} },
41     {tag {f"CONSTRAINT_TYPE"},          value{us "PRIMARY KEY"} },
42     {tag {f"TABLE_NAME"},          value{us "Purchase_order"} },

```

```

1   {tag {f"IS_DEFERRABLE"}, value{us"NO"} },
2   {tag {f"INITIALLY_DEFERRED"}, value{us"NO"} },
3   },
4   {   {tag {f"CONSTRAINT_NAME"}, value{us "P02"} },
5   {tag {f"CONSTRAINT_TYPE"}, value{us"FOREIGN KEY"} },
6   {tag {f"TABLE_NAME"}, value{us"Purchase_order"} },
7   {tag {f"IS_DEFERRABLE"}, value{us"YES"} },
8   {tag {f"INITIALLY_DEFERRED"}, value{us"NO"} },
9   },
10  {   {tag {f"CONSTRAINT_NAME"}, value{us "POI1"} },
11  {tag {f"CONSTRAINT_TYPE"}, value{us"PRIMARY KEY"} },
12  {tag {f"TABLE_NAME"}, value{us"Purchase_order_item"} },
13  {tag {f"IS_DEFERRABLE"}, value{us"NO"} },
14  {tag {f"INITIALLY_DEFERRED"}, value{us"NO"} },
15  },
16  {   {tag {f"CONSTRAINT_NAME"}, value{us "POI2"} },
17  {tag {f"CONSTRAINT_TYPE"}, value{us"FOREIGN KEY"} },
18  {tag {f"TABLE_NAME"}, value{us"Purchase_order_item"} },
19  {tag {f"IS_DEFERRABLE"}, value{us"YES"} },
20  },
21  {   {tag {f"CONSTRAINT_NAME"}, value{us "POI3"} },
22  {tag {f"CONSTRAINT_TYPE"}, value{us"FOREIGN KEY"} },
23  {tag {f"TABLE_NAME"}, value{us"Purchase_order_item"} },
24  {tag {f"IS_DEFERRABLE"}, value{us"YES"} },
25  {tag {f"INITIALLY_DEFERRED"}, value{us"NO"} },
26  },
27  },
28  {tablename "SQL_KEY_COLUMN_USAGE"          --begin table in C.3.9
29  {   {tag {f"CONSTRAINT_NAME"}, value{us "PR1"} },
30  {tag {f"TABLE_NAME"}, value{us"Product"} },
31  {tag {f"COLUMN_NAME"}, value{us"Product_code"} },
32  {tag {f"ORDINAL_POSITION"}, value{us"1"} },
33  },
34  {   {tag {f"CONSTRAINT_NAME"}, value{us "SU1"} },
35  {tag {f"TABLE_NAME"}, value{us"Supplier"} },
36  {tag {f"COLUMN_NAME"}, value{us"Supplier_code"} },
37  {tag {f"ORDINAL_POSITION"}, value{us"1"} },
38  },
39  {   {tag {f"CONSTRAINT_NAME"}, value{us "P01"} },
40  {tag {f"TABLE_NAME"}, value{us"Purchase_order"} },
41  {tag {f"COLUMN_NAME"}, value{us"Purchase_order_no"} },
42  {tag {f"ORDINAL_POSITION"}, value{us"1"} },

```

```

1      },
2      { tag {f"CONSTRAINT_NAME"}, value{us "P02"} },
3      { tag {f"TABLE_NAME"}, value{us "Purchase_order"} },
4      { tag {f"COLUMN_NAME"}, value{us "Supplier_code"} },
5      { tag {f"ORDINAL_POSITION"}, value{us "1"} },
6      },
7      { tag {f"CONSTRAINT_NAME"}, value{us "POI1"} },
8      { tag {f"TABLE_NAME"}, value{us "Purchase_order_item"} },
9      { tag {f"COLUMN_NAME"}, value{us "Purchase_order_no"} },
10     { tag {f"ORDINAL_POSITION"}, value{us "1"} },
11     },
12     { tag {f"CONSTRAINT_NAME"}, value{us "POI1"} },
13     { tag {f"TABLE_NAME"}, value{us "Purchase_order_item"} },
14     { tag {f"COLUMN_NAME"}, value{us "Item_no"} },
15     { tag {f"ORDINAL_POSITION"}, value{us "2"} }
16     },
17     { tag {f"CONSTRAINT_NAME"}, value{us "POI3"} },
18     { tag {f"TABLE_NAME"}, value{us "Purchase_order_item"} },
19     { tag {f"COLUMN_NAME"}, value{us "Product_code"} },
20     { tag {f"ORDINAL_POSITION"}, value{us "1"}}
21     },
22     }
23     {tablename "SQL_REFERENTIAL_CONSTRAINT" --begin table in C. 3. 10
24     { tag {f"CONSTRAINT_NAME"}, value{us "P02"}},
25     { tag {f"MATCH_OPTION"}, value{us "FULL"}},
26     { tag {f"UPDATE_RULE"}, value{us "SET DEFAULT"}},
27     { tag {f"DELETE_RULE"}, value{us "SET DEFAULT"} },
28     },
29     { tag {f"CONSTRAINT_NAME"}, value{us "POI2"}},
30     { tag {f"MATCH_OPTION"}, value{us "FULL"}},
31     { tag {f"UPDATE_RULE"}, value{us "SET DEFAULT"}},
32     { tag {f"DELETE_RULE"}, value{us "CASCADE"}},
33     },
34     { tag {f"CONSTRAINT_NAME"}, value{us "POI3"}},
35     { tag {f"MATCH_OPTION"}, value{us "FULL"}},
36     { tag {f"UPDATE_RULE"}, value{us "SET DEFAULT"}},
37     { tag {f"DELETE_RULE"}, value{us "SET DEFAULT"}},
38     },
39     }
40     {tablename "SQL_CHARACTER_SET" --begin table in C. 3. 15
41     { tag {f"CHARACTER_SET_NAME"}, value{us "Norwegian Riksmål"}},
42     { tag {f"FORM_OF_USE"}, value{us "Standard"}},

```

```

1   {tag {f"NO_OF_CHARACTERS"},      value{us"29"}},
2   },
3   }
4   {tablename "SQL_USER"            --begin table in C. 3. 17
5   {   {tag {f"USER_NAME"},          value{us "Z6AWPQ"}},
6     },
7     {   {tag {f"USER_NAME"},          value{us "YTWENS"}},
8       },
9     {   {tag {f"USER_NAME"},          value{us "0WA5P"}},
10      },
11    }
12  },
13  {tablename "SQL_TABLE_PRIVILEGE"  --begin table in C. 3. 18
14  {   {tag {f"GRANTOR"},              value{us "Z6AWPQ"}},
15    {tag {f"GRANTEE"},                value{us "YTWEN5"}},
16    {tag {f"TABLE_NAME"},              value{us "Product"}},
17    {tag {f"PRIVILEGE_TYPE"},          value{us"SELECT"} },
18    {tag {f"IS_GRANTABLE"},            value{us"NO"} },
19  },
20  {   {tag {f"GRANTOR"},                value{us "Z6AWPQ"}},
21    {tag {f"GRANTEE"},                  value{us "0WA5P"}},
22    {tag {f"TABLE_NAME"},                value{us "Supplier"}},
23    {tag {f"PRIVILEGE_TYPE"},            value{us"SELECT"} },
24    {tag {f"IS_GRANTABLE"},              value{us"NO"} },
25  },
26  {   {tag {f"GRANTOR"},                value{us "Z6AWPQ"}},
27    {tag {f"GRANTEE"},                  value{us "YTWEN5"}},
28    {tag {f"TABLE_NAME"},                value{us "Purchase_order"}},
29    {tag {f"PRIVILEGE_TYPE"},            value{us"SELECT"} },
30    {tag {f"IS_GRANTABLE"},              value{us"NO"} },
31  },
32  {   {tag {f"GRANTOR"},                value{us "Z6AWPQ"}},
33    {tag {f"GRANTEE"},                  value{us "0WA5P"}},
34    {tag {f"TABLE_NAME"},                value{us "Purchase_order_item"}},
35    {tag {f"PRIVILEGE_TYPE"},            value{us"SELECT"} },
36    {tag {f"IS_GRANTABLE"},              value{us"NO"} },
37  },
38  }
39  }
40  --end of SQL-definition
=====

```

1 C.5.4 SQL-data expressed in ASN.1

2 This part consists of each SQL application table and its content for each of the four application tables in C.4 and
 3 the associated content. The explanatory notes at the right hand end of each line are shown in full for the first table,
 4 and omitted for the third and subsequent tables.

```

5  =====
6  {          --part 3 of 3
7  {title      ("ISO SQL- data")
8  Sql data
9  {          --begin table set
10 {tablename "Product"          --begin table in C. 4. 1
11 {  --begin row 1
12 {tag {f"Product_code"},          value{us "FX198A"}},          -- col 1
13 {tag {f"Product_description"}, value{us "østehøvel"} -- col 2
14 },  --end column last
15 },  --end row 1, not last
16 {          --begin row 2
17 {tag {f"Product_code"},          value{us "FV236J"}},          col 1
18 {tag {f"Product_description"}, value{us "Brødkniv"} -- col 2
19 },  --end column last
20 },  --end row 2, not last
21 {          --begin row 3
22 {tag {f"Product_code"},          value{us "FE437P"}},          -- col 1
23 {tag {f"Product_description"}, value{us "Tallerken"}-- col 2
24 },  --end column last
25 },  --end row 3, not last
26
27 {          --begin row 4
28 {tag {f"Product_code"},          value{us "TN328D"}},          col 1
29 {tag {f"Product_description"}, value{us "Blåbør"} -- col 2
30 },  --end column last
31 },  --end row 4, last
32 }  --end table
33 {tablename "Supplier"          --begin table in C. 4. 2
34 {  {tag {f"Supplier_code"},          value{us "169235"}},
35 {tag {f"Supplier_description"}, value{us "Ol sen"}},
36 },
37 {  {tag {f"Supplier_code"},          value{us "174763"}},
38 {tag {f"Supplier_description"}, value{us "Jensen"}},
39 },
40 {  {tag {f"Supplier_code"},          value{us "549362"}},
41 {tag {f"Supplier_description"}, value{us "Hansen"}},
42 },

```

```

1      {   {tag {f"Supplier_code"},           value{us "589267"}},
2      {tag {f"Supplier_description"},       value{us "Petersen"}},
3      },
4      }
5      {tablename "Purchase_order"           --begin table in C.4.3
6      {   {tag {f"Purchase_order-no"},      value{us "658"}},
7      {tag {f"Supplier_code"},             value{us "169235"}},
8      {tag {f>Date_of_order"},            value{us "23. 12. 1999"}},
9      },
10     {   {tag {f"Purchase_order-no"},      value{us "659"}},
11     {tag {f"Supplier_code"},             value{us "169235"}},
12     {tag {f>Date_of_order"},            value{us "23. 12. 1999"}},
13     },
14     }
15     {tablename "Purchase_order_item"       --begin table in C.4.4
16     {   {tag {f"Purchase_order-no"},      value{us "658"}},
17     {tag {f"Item_no"},                   value{us "1"}},
18     {tag {f"Product_code"},              value{us "TN328D"}},
19     {tag {f"No_ordered"},                 value{us "73"}},
20     },
21     {   {tag {f"Purchase_order-no"},      value{us "658"}},
22     {tag {f"Item_no"},                   value{us "2"}},
23     {tag {f"Product_code"},              value{us "FV236J"}},
24     {tag {f"No_ordered"},                 value{us "29"}},
25     },
26     {   {tag {f"Purchase_order-no"},      value{us "659"}},
27     {tag {f"Item_no"},                   value{us "1"}},
28     {tag {f"Product_code"},              value{us "TN328D"}},
29     {tag {f"No_ordered"},                 value{us "171"}},
30     },
31     {   {tag {f"Purchase_order-no"},      value{us "659"}},
32     {tag {f"Item_no"},                   value{us "2"}},
33     {tag {f"Product_code"},              value{us "FE437P"}},
34     {tag {f"No_ordered"},                 value{us "143"}},
35     },
36     }
37     }
38     =====

```

1 **Editor's notes**

2 **N01- Editor's note This foreword will be removed in the final document vi**

3 **N02- Editor's note This foreword will be removed in the final document vii**

4 **N03- Editor's note This will be replaced at the DIS stage by the ITTF approved text viii**

5 **1 Scope..... 1**

6 **N04- Modified because of the addition of the "Select for Export Service".Maybe the sentence needs**
 7 **to be removed. 1**

8 **2 Conformance..... 1**

9 **N05- Editor's note From an SQL point of view, it needs to be considered whether it is possible to**
 10 **define a level of conformance which is independent of ASN.1 2**

11 **3 Normative reference(s) 2**

12 **4 Term(s) and definition(s) 3**

13 **N06- Editor's note: They are repeated here verify their applicability and to facilitate review of this**
 14 **document 4**

15 **N07- Editor's note: They are repeated here to facilitate review of this document..... 4**

16 **N08- Editor's note : This process could be used from another kind of environment such as a**
 17 **spreadsheet. This capability has not been referenced further in the current version. The full implications of**
 18 **this remain to be analyzed..... 6**

19 **N09- Editor's note: This process could be used into another kind of environment such as a**
 20 **spreadsheet. This capability has not been referenced further in the current version. The full implications of**
 21 **this remain to be analyzed..... 6**

22 **5 Symbols (and abbreviated terms) 6**

23 **N10- Made consistent with IRDS part (Facility replaces File)..... 7**

24 **6 Concept and Facilities 7**

25 **7 Specification of the transfer file 9**

26 **N11- Editor's Note: This has been made consistent with the IRDS document. However the rest of**
 27 **this document discusses three possible codesets. Is that obsolete and need to be removed, or is it**
 28 **because of some interaction with the SQL standard? In any case Section 7.1 and 8.2.2 need to be made**
 29 **consistent. 9**

30 **N12- Editor'snote Replace ?? above..... 10**

31 **8 Transfer file Structure..... 11**

32 **N13- Editor's Notes: The syntax for the header should be BNF, and the encoding CLEAR. To satisfy**
 33 **the design objective of standardized headers across transfer file formats, the header itself should not be**
 34 **ASN.1, but indicate that the following components are ASN.1..... 11**

1	N14-	Editor's notes: Insert here the additional BNF to fully specify the header in terms of BNF, if	
2		that option is selected.	13
3	N15-	Editor's Note: The previous document had a version number for the standard, it has been	
4		removed, as the year indicator act the same way.	13
5	N16-	Editor's note Clarification of intended use is needed here.	13
6	N17-	Editor's note. The context in which <PublisherName> is unique needs to be clarified. It is not	
7		clear how such names are intended to be allocated, although there are various options. The above note is	
8		the minimum needed.	13
9	N18-	Editor's note <SQL name> is not a term used in 9075. However, it is introduced in 10728. It	
10		may not be an appropriate term to use in 13239-2.	16
11	N19-	Editor's note The phrase "or zero" should be deleted from this note. If data values of zero are	
12		not included, the importer will not know whether the value is null or zero.	17
13	N20-	Editor's note It needs to be clarified whether visible string can contain all possible column	
14		name characters.	17
15	N21-	Editor's Note: What happens when columns names are not used	18
16	N22-	Editor's notes: For the following option, unambiguous and computable rules must be prepared	
17		to specify what the sequence of values is, or it must be specified that the sequence is implementor-	
18		defined. This is applicable both in the case of SQL-definition tables, and SQL-data tables.....	18
19	N23-	Editor's notes: For the following options, unambiguous and computable rules must be	
20		prepared to derive the column identifier, This is applicable both in the case of SQL-definition tables, and	
21		SQL-data tables	18
22	9	Specification of export/import processes.....	19
23	N24-	Editor's note Some of the fields in the SQL Header are not precisely compatible with the IRDS	
24		E/I Header. Should we make them so ?	21
25	N25-	Editor's notes: This constraint on TfTableRowTags needs to be articulated.	24
26	N26-	Editor'snote does the complete table for the previous procedure apply ?	26
27	N27-	Editors Editor's note There is no entry in 9075 for invalid table name (for example) and it needs	
28		to be clarified how exceptions such as this are to be handled.....	30
29	N28-	Editor'snote The following are potential additional error code valuyes:.....	30
30	N29-	Editor's note Clarification is needed of what is <appropriate>.....	30
31	N30-	Editor'snote The following two rules need amending in the light of the above new rules.	31
32	10	Notes and Tables	33
33	Annex A (Informative)	Conventions mandated by use of ASN.1.....	34
34	N31-	Editor's note This annex must either be made consistent with the main body of the standard or	
35		else removed.....	34
36	Annex B (normative)	ASN.1 Definition of the Transfer Format	35

1 **N32- Editor's note This annex is currently: a) Not harmonized with the rest of the document; b) Not**
2 **harmonized with the use of ASN.1 in the recent RDA document; c) Not harmonized with the recent**
3 **improvements to ASN.1 for "tables"..... 35**

4 **Annex C (informative) Sample Transfer file Format..... 42**

5 **N33- Editor's note The actual ASN.1 value notation needs aligning with the corrections to the type**
6 **notation in the body of the standard..... 48**

7

1 Control Fields (Show field codes to adjust)

	ISO/IEC	
	ISO/IEC J	
TC #	1	
SC #	32	
WG #	3	
Secretariat SC	SCC-Canada	
Doc Type	International Standard	
Doc Pub year		
Doc Edition #	1	
Doc Subtype	Not applicable	
Language	E	
Stage	(30) Committee	
Title (for Stage)	COMMITTEE DRAFT	
ISO Number	CD 13238-2	
English Title	Information Technology — Data Management Export/Import — Part 2: SQL Export/Import	
French Title	Technologies de l'information — Échange (export/import) de données — Partie 1: Export/Import pour SQL	
SC Doc Number	0019	
WG Doc Number	2116	
Document Date	1997-11-10	
Version	LHR05 -- 97.09.15	
Stamp	97.11.12 12:41	

1 **Status Fields**

2

Title:	DMEI SQL Export/Import	
Subject:	Output of London July 97 - Cd Circulation Sept	
Keywords:	ISO Template Version 3.2 1997-07-10 JBEV03	
Comments:		
Version:	LHR05 -- 97.09.15	
Filename:	32N0019_SQL_EI_323N2116_V10.doc	
Template:	DCISOV03.dot	
Original Edit.:	TW Olle	
Current Edit.:	Jean Bérubé	
Create Date:	97.11.12 12:43	
Save Date:	97.11.12 12:43	
Print Date:	97.11.12 12:45	
Stamp	97.11.12 12:43	
Modification:	2	
Edit Time:	0.03 hr.	
Num. char:	92172	
Num. words	15945	
Num. pages	69	

3