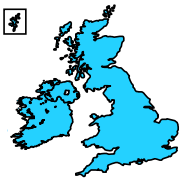
SQL:1999 and Recent Developments in SQL Standardisation

by Hugh Darwen
IBM United Kingdom Limited
Hugh_Darwen@uk.ibm.com



22 May, 2001

(Cambridge University)

The Parts of SQL

Part	Title	Published	Status
1	Framework	99, 02	IS, CD
2	Foundation	86, 89, 92, 99, 02	IS, CD
3	Call-Level Interface (CLI)	95, 99, 02	IS, CD
4	Persistent Stored Modules (PSM)	96, 99, 02	IS, CD
5	Host Language Bindings	92, 99, withdrawn in 2002	IS
6	(unused part number)		
7	Temporal	?	WD
8	(unused part number)		
9	Management of External Data (MED)	2000/1, 2002	FDIS, CD
10	Object Language Bindings (OLB) (embedded SQL for Java)	2000, 2002	FDIS, CD
11	Schemata	2002	CD
12	Replication	?	none
13	Java routines and types	?	none
14	XML stuff	?	none

+ OLAP "amendment", published in 2000

SQL/Foundation

1. Data definition and manipulation
   - new builtin data types
   - user-defined data types
   - CREATE TABLE ... LIKE ...
   - SENSITIVE cursors, WITH HOLD cursors
2. Functions and operators
   - User-defined "routines"
   - Recursive table expressions
   - Enhanced view updatability
   - New quantifiers
   - SIMILAR condition
   - DISTINCT condition
   - Cross tabulation
3. ?Object Support
4. Integrity
   - Referential action RESTRICT
   - Triggers
5. Security
   - User-defined "roles"
   - Column-specific SELECT privileges
6. Transactions
   - Transaction savepoints

New Built-In Data Types

- BOOLEAN
  - only two values, NULL standing for "unknown"
  - ALL and SOME aggregate functions
- BLOB and CLOB
  - strings, including much larger ones than you can have with BIT and CHARACTER
  - special "locator" mechanism for avoiding costly movement of data between client and server
- ROW (constructor)
  - so you can have nested rows
- ARRAY (constructor)
  - the only "collection types"
  - nested tables et al. deferred to SQL4
- REF (constructor)
  - for what some might call oids

User-Defined Data Types

- "Structured" types
  - subtyping, single inheritance only
  - representations in terms of "attributes", inherited by subtypes
  - attribute reference ("observer") is shorthand for method invocation, and can also be assigned to ("mutator")
- "Distinct" types
  - a data type name with an underlying type
  - e.g. SHOE_SIZE and INTEGER
- "Domains" as in SQL:1992
  - not data types at all, really

Recursive Table Expressions

```
WITH  RECURSIVE ANCESTOR_OF AS

  ( SELECT PARENT AS ANCESTOR,
           CHILD AS DESCENDANT
    FROM PARENT_OF
    UNION
    SELECT P.PARENT AS ANCESTOR,
           A.DESCENDANT
    FROM   PARENT_OF P,
           ANCESTOR_OF A
    WHERE P.CHILD = A.ANCESTOR )

SELECT * FROM ANCESTOR_OF
```

★ WITH is *very* useful in its own right, even when you don't want to recurse

Cross Tabulation

```
SELECT AREA, BRANCH, PRODUCT
       SUM(SALES) AS SALES
FROM   SALES_HISTORY
GROUP BY ROLLUP (AREA, BRANCH,
                 PRODUCT );
```

Generates a whole load of nulls!

```
SELECT AREA, BRANCH, PRODUCT
       SUM(SALES) AS SALES
FROM   SALES_HISTORY
GROUP BY CUBE (AREA, BRANCH,
               PRODUCT );
```

Generates nulls all over the place!!

Triggers

```
CREATE TRIGGER ADD_TO_HISTORY
  AFTER DELETE ON WORKS_ON
  REFERENCING OLD AS W
  INSERT INTO WORKED_ON
    (EMPNO, PROJECTNO, FROM, TO)
    VALUES ( W.EMPNO,
             W.PROJECTNO,
             W.FROM,
              CURRENT_DATE )
```

(Maintains history of project assignments)

Persistent Stored Modules

- Published in 1996 as SQL:1992 addendum
- Declaration ("creation") of user-defined "routines" (moved to Foundation in SQL3)
- And a programming language to write them in
  - IF ... THEN ... ELSE ...
  - WHILE ...
  - LOOP
  - FOR every row of some table
  - BEGIN statement-list END
    - local variables (of SQL data types)
    - exception handlers
    - condition names

User-defined Routines

- Signature + properties + body
- + returned type where applicable
- Body can be in any standard language
  - including SQL

Kinds of Routine

- Functions
  - "input" parameters only
  - invocation returns a value
  - "Static dispatch" only
  - overloading supported but no overriding
  - "unselfish"
  - definition not bundled with type definition
- Methods
  - "input" parameters only
  - invocation returns a value
  - "Dynamic dispatch" available
  - overriding supported
  - "selfish"
  - definition bundled with type definition
  - 1st parameter implied by containing type
- Procedures
  - "output" and "inout" parameters supported
  - invocation does not return a value
  - no overloading (or overriding)
  - definition not bundled with type definition

Special-purpose UDFs

- User-defined casts
  - implicitly invoked on CAST from DT1 to DT2
- User-defined transforms
  - Implicitly invoked on data transfer to and from host language variables (no more impedance mismatch!)

?Object Support

- Table defns based on structured types
- Subtables (single inheritance)
- System-generated keys (i.e. reference values, or oids) via REF data types
- dereferencing operators

```
CREATE TYPE Department
    ( DEPTNO CHAR(5),
      LOCATION VARCHAR(30),
      BUDGET MONEY );

CREATE TABLE DEPT OF Department
   ( DEPT_OID REF(Department),
     UNIQUE DEPTNO );

CREATE TABLE EMP ( ...,
  DEPT_REF REF(Department),
  SCOPE FOR DEPT_REF IS DEPT );

SELECT ENAME, DEPT_REF->LOCATION
FROM    EMP
```

Populating REFs

```
INSERT INTO DEPT VALUES (
    'D0005', 'Warwick',
    MONEY(505000) );
```

Note: DEPT_OID value system-generated and different from all other DEPT_OID values in DEPT.

```
INSERT INTO EMP VALUES (
    'E0123', 'Hugh Darwen',
    MONEY(500000),
    ( SELECT DEPT_OID
      FROM    DEPT
      WHERE DEPTNO = 'D0005' ) );
```

I wonder what CJD is going to say about all this!

(*Added later:* he's said it! Foundation for Future Database Systems, Addison-Wesley. ISBN 0-201-70928-7)

Subtables

Given a structured type T1 and a structured type T2 that is a subtype of T1, you can create a table TAB1 of type T1 and then create a table of type T2 "under" TAB1. E.g.:

```
CREATE TABLE EMP OF Employee;

CREATE TABLE MGR OF Manager
    UNDER EMP;
```

Every row inserted into MGR shows up in EMP too, but rows inserted into EMP do not show up in MGR.

But there's no "shorthand" for promoting and demoting employees! (yet)

Conformance and Certification

- "Core" + packages
- E.g.:
  - Basic Object Support
  - Enhanced Object Support
  - OLAP facilities (CUBE etc.)
  - Active database support (triggers)
  - Enhanced datetime facilities
  - Enhanced integrity management
  - PSM
  - CLI

To Read All About It ...

Current FTP site for working drafts is

ftp: //sqlstandards.org/SC32/WG3/Progression_Documents