SQL Standardi

Relacioni model baza podataka postajao je osamdesetih godina 20. veka polako ali sigurno industrijski standard. Mada je SQL bio široko prepoznati jezik baza podataka, problem je predstavljao porast razlika između implementacija raznih proizvođača. Neka vrsta standarda postala je neophodna.

Organizacija sa neobično dugačkim imenom - American National Standards Institute National Committee on Information Technology Standards H2 Technical Committee on Database (ANSI NCITS H2 TCD) dobila je zadatak da standardizuje relacioni model 1982 godine. Projekat je u početku bio baziran na IBM SQL/DS specifikaciji, i u jednom periodu je striktno pratio razvoj IBM DB2. Godine 1984. predlog standarda je redizajniran tako da je postao opštiji i dopustio veću raznolikost proizvođača sistema baza podataka.

SQL je prvi put standardizovan ***1986***. godine pod nazivom "Database LanguageSQL" (Američki nacionalni institut za standardizaciju, X3.135) odnosno ***1987***. godine (Međunarodna organizacija za standardizaciju, ISO). Ovaj ANSI/ISO standard usvojila je i organizacija Federalnih standarda za obardu informacija američke vlade (Federal Information Processing Standard, *FIPS*). Standard je imao manje od 100 strana i uključivao je malo svojstava – jednostavne upite, ažuriranja i definicije tabela, tabelarni model podataka, nezavisan od jezika, definiciju sheme, poglede i kursore za sumeđe (engl. interface) ka slogovno orijentisanim programskim jezicima.

Revidirani standard, poznat kao *SQL89* ili *SQL1*, objavljen je 1989. godine.

SQL89 (SQL1)

Zbog suprotstavljenih interesa komercijalnih proizvođača, veliki deo SQL89 (ili SQL1) standarda ostavljen je namerno nepotpun, i mnoga svojstva su označena kao "definisana implementacijom". Standard je imao oko 120 strana i dodao je samo nekoliko vrsta ograničenja za očuvanje integriteta uskladištenih podataka. Obezbeđuje referencijalni integritet i ugnježdenje SQL-a u programske jezike Ada, C, COBOL, FORTRAN, Pascal i PL/I.

SQL92 (SQL2)

Da bi se standard pojačao, ANSI kometiet je revidirao prethodni rad standardom *SQL92* koji je bio baziran na SQL89 i ratifikovan 1992. (nazvan i *SQL2*). Ovaj standard je razrešio veći broj slabosti standarda SQL89 i postavio konceptualna SQL svojstva koja su u to vreme prevazilazila svojstva postojećih RSUBP implementacija. Standard SQL92 bio je približno pet puta veći od svog prethodnika (ima oko 600 strana) i on poboljšava ortogonalnost jezika dopuštajući upotrebu izraza na svakom mestu gde se očekuju tabele ili skalarne vrednosti (npr. SELECT, FROM linija). SQL dodaje nova svojstva kao što su tipovi podataka *date* i *time*, skupovne operatore UNION, INTERSECT, EXCEPT, standardne tabele kataloga za smeštanje metapodataka, svojstva evolucije sheme kroz ALTER TABLE, itd. Neka od rešenja ovog standarda odnose se na:

dinamički SQL kroz PREPARE i EXECUTE iskaze
spoljašnje spajanje
kaskadno ažuriranje i brisanje

privremene tabele
skupovne operacije unije, preseka i razlike
definiciju domena u shemi
nove ugrađene tipove podataka
nivoe konzistentnosti transakcija
odloženu proveru uslova integriteta
SQL dijagnostiku, itd.

Na primer, lista izvedenih tabela u FROM liniji:

```
SELECT P_SIF, I_SIF
        FROM P, (SELECT *
                        FROM I
                        WHERE STATUS > 10) AS DOBRIIZD
        WHERE P.DRZAVA = DOBRIIZD.DRZAVA
```

SELECT upitni blok u SELECT liniji, npr.

```
        SELECT I.I_SIF, (SELECT SUM(KI.TIRAZ)
                FROM KI
                WHERE KI.I_SIF = I.I_SIF ) AS UK_TIRAZ
        FROM I
```

Prošireni logički izrazi, npr.

(p) IS [NOT] TRUE ((p) je (nije) tačno)
(p) IS [NOT] FALSE ((p) je (nije) netačno)
(p) IS [NOT] UNKNOWN ((p) je (nije) nepoznato)

Dekartov proizvod, prošireno prirodno, slobodno spajanje, unija - operacija oblika:

tabela-1 CROSS JOIN |

[NATURAL] [INNER|LEFT|RIGHT|FULL|UNION] JOIN

tabela-2

[ON *logički-izraz*]

Na primer,

```
        SELECT K.K_SIF, SUM(TIRAZ) AS TIR
        FROM (K NATURAL LEFT JOIN KI)
        GROUP BY K.K_SIF
```

Za kreiranje domena koristi se iskaz

CREATE DOMAIN *naziv-domena* [AS] *tip-podataka*
[*podrazumevana-definicija*]
[*lista-definicija-ogranicenja-domena*]

Opšti uslov integriteta

CREATE ASSERTION *ime-pravila* CHECK(*uslov*), na primer

```
CREATE ASSERTION KI1 CHECK
    (NOT EXISTS ( SELECT * FROM I, KI
    WHERE I.STATUS < 20 AND
    I.I_SIF = KI.I_SIF AND KI.TIRAZ > 5000))
```

Zbog obimnosti novog standarda autori su definisali tri nivoa saglasnosti sa SQL92:

Ulazni nivo (*Entry-level conformance*) – samo blaga poboljšanja/proširenja standarda SQL89, na primer: iskazi za definisanje sheme, jezik manipulisanja podacima, referencijalni integritet, provera uslova integriteta, ugnježdenje SQL-a u sedam različitih programskih jezika, direktno izvršenje iskaza manipulisanja podacima, SQLSTATE parametar, (pre)imenovanje kolona u SELECT listi, opcija pri kreiranju pogleda WITH CHECK OPTION;

Srednji nivo (*Intermediate-level conformance*) – skup značajnih poboljšanja, uključujući imenovanje ograničenja (constraints), stringovi promenljive dužine, nacionalni skupovi karaktera, izrazi *case* i *cast*, ugrađeni operator spajanja, dinamički SQL, izmena tabela (alter), definisanje transakcija I nivioi izolovanosti, korišćenje podupita u pogledima koji se mogu ažurirati, skupovni operatori (UNION, EXCEPT, INTERSECT).

Puni nivo saglasnosti (*Full-level conformance*) uključuje neka suštinski napredna svojstva, kao što su odložena provera ograničenja, tvrđenja (assertions), privremene lokalne tabele, privilegije nad skupovima karaktera i domenima, proširenu podršku tipovima datuma/vremena, ažuriranje i brisanje iz tabele nad kojom je zadat i logički izraz WHERE linije, kaskadno ažuriranje, niske bitova kao tipove podataka, tvrđenje ograničenja.itd.

SQL99 (SQL3)

Između 1992. i 1999. objavljena su dva specijalizovana proširenja SQL-92, nazvana Call Level Interface (**CLI**) i Persistent Stored Modules (**PSM**). CLI definiše skup funkcija koje se mogu pozivati iz programa napisanih u jezicima kao što je C, i čijim se pozivima vrši dinamičko povezivajnje (konekcija) sa relacionom bazom podataka i izvršavanje SQL iskaza. PSM proširuje SQL iskazima dodele, kontrolnim iskazima i obradom izuzetaka, što omogućuje implementaciju aplikacija nad bazama podataka u samom SQL-u. Tako SQL postaje izračunljivo kompletan, eliminušući potrebu za ugnježdenjem u programske jezike opšte namene, npr. C. PSM je pokušaj da se standardizuju proceduralna proširenja kao što je PL/SQL RSUBP Oracle ili Transact-SQL SQL-server-a.

Godine 1999. ANSI/ISO objavio je *SQL99* standard (nazvan i SQL:1999 ili *SQL3*). Novi standard je razbijen u nekoliko delova, uključujući CLI i PSM kao deo 3 odnosno 4 (Part 3, 4).

SQL:1999 uključuje značajno novu funkcionalnost kao što su trigeri, veliki objekti, rekurzivni upiti, korisnički definisane funkcije. Značajan naglasak je stavljen na objektno-relacionu funkcionalnost i na nova svojstva za on-line analytic processing (OLAP). Skup svih delova ovog standarda prelazi 2000 strana. Povremeno se dodaju i novi delovi standardu.

Novi standard je razvijan pod zajedničkim vođstvom ANSI i ISO komiteta i uključuje neke naprednije i prethodno zanemarivane oblasti modernih SQL sistema, kao što su koncepti objektno-relacionih sistema, interfejs nivo poziva (call level interfaces, CLI) i upravljanje integritetom. SQL99 je zamenio SQL92 nivoe saglasnosti svojim sopstvenim stepenima

usaglašenosti: osnovni SQL99 (*Core SQL99*) i pojačani SQL99 (*Enhanced SQL99*). Veoma je kompleksan (preko 1,500 strana u odnosu na 120 strana standarda SQL89 i oko 600 strana SQL92).

SQL3 uključuje objekte i kompleksne tipove podataka i sve mehanizme za podršku, unutar relacionih tabela. U SQL se uvode svi bitni principi objektnog programiranja – nasleđivanje, učaurenje i polimorfizam.

SQL99 relaciona (ne-OO) svojstva

Tipovi podataka

SQL:1999 ima četiri nova tipa podataka:

1. LARGE OBJECT tip (LOB), sa varijantama CHARACTER LARGE OBJECT (CLOB) i BINARY LARGE OBJECT (BLOB). LOB objekat ne može da bude primarni ključ (PRIMARY KEY), strani ključ (FOREIGN KEY) i ne može da se koristi u poređenjima osim testova jednakosti ili različitosti, ne može da se koristi u GROUP BY ili ORDER BY liniji. Inače se CLOB tip ponaša slično nisci karaktera.

2. Tip je BOOLEAN sa vrednostima *true*, *false*, i *unknown*.

3,4. Dva nova strukturna (kompozitna) tipa: ARRAY i ROW. Tip ARRAY dopušta da se kolekcije vrednosti dodele atributu (koloni) tabele baze podataka. Na primer,

DANI VARCHAR(10) ARRAY[7] omogućuje smeštanje imena svih sedam dana u jedan atribut (DANI) jedne vrste tabele. Ovo svojstvo narušava normalizovanost relacije (1NF).

Tip ROW dalje narušava 1NF dopuštajući strukturnu vrednost u koloni tabele: na primer,

```
ADDRESS ROW (
                STREET  VARCHAR(50),
                CITY  VARCHAR(30),
                STATE   CHAR(2)  )
```

SQL:1999 uvodi i jednostavne korisnički definisane tipove – DISTINCT TYPES - koji su definisani nad jednostavnim ugrađenim (predefinisanim) SQL tipovima ali se ne mogu "mešati" u istom izrazu bez eksplicitne konverzije operatorom CAST (primer u SQLstandardi.ppt)

Predikati

Od nekoliko novih predikata u SQL:1999, pomenimo predikat **SIMILAR** koji omogućuje UNIX-like zadavanje regularnog izraza, korisno u sravnjivanju obrazaca (pattern matching). Na primer,

```
WHERE NAME SIMILAR TO '(SQL-(86|89|92|99))|(SQL(1|2|3))'
```

Bogatija semantika

SQL99 doprinosi obogaćenju semantike jezika na više načina. Jedan aspekt je proširenje skupa pogleda koji se mogu ažurirati (npr. pogledi koji uključuju podupite) analizom funkcionalnih

zavisnosti i definisanjem pravila ažuriranja baznih tabela (npr. pogledi koji sadrže distinct opciju u SELECT liniji)

Drugi aspekt obogaćenja semantike jeste podrška rekurzivnim upitima.

Rekurzivni upit se sastoji od inicijalnog SELECT uptnog bloka koji izračunava neki preliminarni rezultat, i rekurzivnog SELECT upitnog bloka koji izračunava dodatni rezultat na bazi vrednosti koje su prethodno izračunate. Rekurzivni SELECT blok se izračunava ponovljeno sve dok proizvodi dodatak rezultatu. Rekurzivni upiti su korisni kada se traži optimalni rezultat u nekom prostoru, npr. "Naći najjeftiniju kombinaciju deonica leta od Šangaja do Kopenhagena", ili kada se izračunava neki oblik sastavnice. Na primer, neka je data tabela part_structure sastavnih delova:

MAJOR_P	MINOR_P	QTY
P1	P2	2
P1	P3	4
P2	P3	1
P2	P4	3
P3	P5	9
P4	P5	8
P5	P6	3

Značenje jednog reda je da deo MAJOR_P uključuje u svoj sastav QTY primeraka dela MINOR_P. Ako treba izračunati "sastavnicu" za dati deo, tj. odrediti sve delove koji ulaze u sastav svakog dela (*ili određenog dela, npr. P1*), to se može izraziti rekurzivnim upitom sledećeg oblika:

```
with subpart(major_p, minor_p) as
  (select major_p, minor_p
            from part_structure
      union all
   select subpart.major_p, part_structure.minor_p
   from subpart, part_structure
   where subpart.minor_p=part_structure.major_p
  )
select distinct major_p, minor_p
from subpart;
(where major_p="P1")
```

U sintaksi SQL99,

```
WITH RECURSIVE
          Q1 AS SELECT...FROM...WHERE...,
          Q2 AS SELECT...FROM...WHERE...
SELECT...FROM Q1, Q2 WHERE...
```

```
WITH RECURSIVE subpart(major_p, minor_p) AS
          ( SELECT major_p, minor_p
            FROM part_structure
            WHERE major_p =“P1”
UNION ALL
            SELECT sp.major_p, p.minor_p
```

```
        FROM subpart sp, part_structure p
        WHERE sp.minor_p = p.major_p )

SELECT DISTINCT major_p, minor_p
FROM subpart
```

SQL99 OO svojstva

Strukturni korisnički-definisani tipovi predstavljaju fundamentalno objektno orijentisano svojstvo ovog standarda. Imaju niz karakteristika:

Mogu da imaju jedan ili više atributa proizvoljnih tipova, uključujući i strukturne korisnički definisane tipove;

Svi aspekti ponašanja obezbeđeni su funkcijama, metodama i procedurama

Poređenje vrednosti vrši se korisnički definisanim funkcijama

Mogu da učestvuju u hijerarhijama tipova kao potipovi / nadtipovi.

Na primer,

```
CREATE TYPE emp_type
  UNDER person_type
AS ( EMP_ID      INTEGER,
     SALARY      REAL )
INSTANTIABLE
NOT FINAL
REF ( EMP_ID )
INSTANCE METHOD
  GIVE_RAISE
    ( ABS_OR_PCT  BOOLEAN,
      AMOUNT      REAL )
    RETURNS REAL
```

Metode

U SQL99, metoda je vrsta funkcija sa nekim ograničenjima i nekim specifičnim svojstvima. Neke od razlika između metode i funkcije ogledaju se u sledećem:

Metoda je vezana za jedinstveni korisnički tip (funkcije nisu);

Korisnički tip za koji je metoda vezana je tip prvog (istaknutog) argumenta metode, koji se ne deklariše; ni jedan argument funkcije nije u tom smislu istaknut;

Metode moraju da budu smeštene u istoj shemi kao i tipovi za koje su vezane, funkcije nisu ograničene na jednu shemu;

I funkcije i metode mogu da budu napisane u SQL/PSM ili programskom jeziku, npr. JAVA.

Atributima korisnički definisanih tipova može da se pristupi korišćenjem tačka-notacije ili funkcijske notacije. Standard SQL:1999 podržava obe notacije kao sintaksne varijante iste radnje. Na primer,

WHERE emp.salary > 10000 ili WHERE salary(emp) > 10000

biće korektno ako je emp – atribut (ili promenljiva) nad strukturnim korisnički definisanim tipom koji ima atribut salary, ili ako je definisana funkcija salary sa jednim argumentom tipa emp_type.

Metode su nešto manje fleksibilne od funkcija i koriste "tačka"-notaciju, bar kada se koriste sa istaknutim argumentom. Ako je *salary* metoda koja je vezana za tip employee, koji je deklarisani tip kolone koja se zove emp, onda metoda može da se pozove samo kao:

emp.salary()

Tipizirana tabela

Prava objektna svojstva SQL:1999 uvodi *tipiziranom tabelom* ("typed table"). Takva tabela deklarisana je nad strukturnim korisnički-definisanim tipom, njene vrste su zapravo vrednosti (instance) tog tipa a kolone su izvedene iz atributa tog tipa. Na primer,

CREATE TABLE empls OF emp_type

Sve funkcije, metode i procedure, definisane za tip nad kojim je definisana tabela, sada se primenjuju na vrste tabele.

Svaka vrsta ima jedinstveni identifikator koji se ponaša kao OID (object identifier) u objektnim sistemima. SQL:1999 uvodi novi, **REF** tip, čije su vrednosti ti jedinstveni identifikatori, a mogu da se koriste kao vrednosti atributa drugih tipova ili tabela. Na primer, u nekoj tabeli kolona *manager* može da se definiše kao REF tip nad tabelom zaposlenih (*empls*), tj. kolona sa vrednostima – identifikatorima vrsta tabele empls:

manager REF(emp_type) with scope empls

Tip REF moguće je da se koristi samo nad tipiziranom tabelom. Za pristup atributima strukturnog korisnički-definisanog tipa koristi se notacija praćenja reference, npr.

SELECT manager->last_name

SQL:2003

Standard SQL:2003 se popularno naziva standardom ispravljanja grešaka standarda SQL:1999 (“bugfix release” - revizija standarda SQL:1999), osim u domenu podrške XML-u. Uprkos tome, ovaj standard ima i niz novih rešenja, a neka od njih su:

Novi tipovi podataka

Uklonjeni su BIT i BIT VARYING tipovi zbog nedostatka podrške u implementacijama, a dodata su tri nova tipa: BIGINT, MULTISET, XML. Tip MULTISET je sličan tipu ARRAY, sa bilo kojim SQL-podržanim tipom elemenata, ali bez uređenja, npr.

MULTISET[1, 2, 3, 4] ili MULTISET(SELECT grades FROM courses)

Nad tipom MULTISET definisane su tri agregatne funkcije – COLLECT (za kreiranje multiset-vrednosti od vrednosti u koloni-argumentu), FUSION (za kreiranje multiset koji unira multiset-vrednosti u koloni-argumentu), INTERSECTION (za kreiranje multiset vrednosti od elemenata iz preseka multiset-vrednosti iz kolone argumenta) na primer, za sledeću tabelu i upit:

FRIEND	HOBBIES
'John'	MULTISET['READING', 'POP-MUSIC', 'RUNNING']
'Susan'	MULTISET['MOVIES', 'OPERA', 'READING']
'James'	MULTISET['MOVIES', 'READING']

SELECT COLLECT(FRIEND) AS ALL_FRIENDS, FUSION(HOBBIES) AS

ALL_HOBBIES, INTERSECTION(HOBBIES) AS COMMON_HOBBIES FROM FRIENDS

Kao rezultat dobija se tabela sa multiset kolonama i jednom vrstom:

ALL_ FRIENDS	ALL_HOBBIES	COMMON_ HOBBIES
MULTISET ['John', 'Susan', 'James']	MULTISET ['READING', 'READING', 'READING', 'POP-MUSIC', 'RUNNING', 'OPERA', 'MOVIES', 'MOVIES']	MULTISET ['READING']

Primer kreiranja tabele sa kolonom MULTISET tipa:

- CREATE TABLE logins
  (session_id INT NOT NULL PRIMARY KEY,
  successful BOOLEAN NOT NULL,
  uid INT,
  attempts ROW(VARCHAR(128),VARCHAR(128)) MULTISET);

- INSERT INTO logins VALUES(1000,true,0,
  MULTISET(
  ROW('root','31337'),
  ROW('scott','tiger'),
  ROW('root','beer')));

- INSERT INTO logins VALUES
  (1001,false,0,MULTISET(SELECT ROW(name,password) FROM bogus_accounts));

Proširenja SQL-rutina

Definisane su funkcije koje proizvode tabele, a mogu da budu zadate nekim ne-SQL jezikom ili SQL izrazom. Na primer, funkcija *weather*, napisana u C-u, proizvodi skup vrsta o vremenskim podacima, koji se smešta u tabelu opisane strukture:

```
CREATE FUNCTION weather()
RETURNS TABLE (
        CITY VARCHAR(25),
        TEMP_IN_F INTEGER,
        HUMIDITY INTEGER,
        WIND VARCHAR(5),
        FORECAST CHAR(25) )
NO SQL
LANGUAGE C
EXTERNAL
```

ili, funkcija DEPTEMPS realizovana u SQL-u:

```
CREATE FUNCTION DEPTEMPS (DEPTNO CHAR(3))
RETURNS TABLE (
        EMPNO CHAR(6),
        LNAME VARCHAR(15),
        FNAME VARCHAR(12))
LANGUAGE SQL
READS SQL DATA
RETURN TABLE (
        SELECT EMPNO, LASTNAME, FIRSTNME
        FROM EMPLOYEE
        WHERE EMPLOYEE.WORKDEPT = DEPTEMPS.DEPTNO)
```

Obraćanju ovako kreiranoj tabeli prethodi rezervisana reč TABLE npr.

```
SELECT W.CITY, W.TEMP_IN_F, W.FORECAST
FROM TABLE(weather()) AS W
WHERE W.TEMP_IN_F > 65
```

MERGE iskaz

Novi MERGE iskaz je, pored INSERT, UPDATE, DELETE iskaza, još jedan iskaz ažuriranja u SQL-u. On omogućuje dodavanje tabele promena glavnoj – "master" tabeli, tj. ažuriranje izmenjenih i dodavanje novih podataka. Na primer, bazna tabela INVENTORY,

Table 1 — INVENTORY table

PARTNUM	DESCRIPTION	QUANTITY
1	Cool Part	10
2	Another Cool Part	15
3	Really Cool Part	20

treba da se ažurira promenama koje su zapamćene u tabeli SHIPMENT:

Table 2 — SHIPMENT table

PARTNUM	DESCRIPTION	QUANTITY
2	Another Cool Part	5
4	Yet Another Cool Part	15
1	Cool Part	10

Iskaz

```
MERGE INTO INVENTORY AS INV
USING (SELECT PARTNUM,DESCRIPTION,QUANTITY FROM SHIPMENT) AS SH
ON (INV.PARTNUM = SH.PARTNUM)
WHEN MATCHED THEN UPDATE
        SET QUANTITY = INV.QUANTITY + SH.QUANTITY
WHEN NOT MATCHED THEN INSERT
(PARTNUM, DESCRIPTION, QUANTITY)
 VALUES (SH.PARTNUM, SH.DESCRIPTION, SH.QUANTITY)
```

proizvodi rezultat

Table 3 — INVENTORY table after MERGE

PARTNUM	DESCRIPTION	QUANTITY
1	Cool Part	20
2	Another Cool Part	20
3	Really Cool Part	20
4	Yet Another Cool Part	15

OLAP

Online analytic processing (OLAP) se koristi u poslovanju za analizu velikih količina podataka da bi se otkrile činjenice i trendovi koji mogu da utiču na poslovne odluke. GROUP BY operator i agregatne funkcije (sum, avg, itd.) ranog SQL-a predstavljejuu primitivne oblike OLAP funkcionalnosti, koja se značajno proširuje kasnijim verzijama jezika.

Na primer, ROLLUP operator omogućuje da upit primeni agregatne funkcije na više nivoa (npr. grad, država). Operator CUBE omogućuje agregiranje podataka po više dimenzija (npr. datum, mesto, kategorija) u jednom upitu.

Multimedija

Godine 2000, SQL Standard je proširen posebnim ali blisko povezanim standardom ‘‘SQLMultimedia and Application Packages’’ (ISO/IEC 13249:2000), koji se često naziva SQL/MM. Ovaj standard koristi objektno-relaciona svojstva uvedena standardom SQL:1999 za definisanje specijalizovanih tipova podataka i metoda za tekst, slike i prostorne podatke.

XML svojstva - SQL:2003 – SQL/XML

XML je format za razmenu podataka kome raste popularnost zato što omogućuje mešanje metapodataka (tagova, obeležja) sa podacima, čime se postiže da podaci sami sebe opisuju. Popularnost XML-a dovela je do zahteva da se XML podaci uskladište u relacionim bazama podataka i da se podaci konvertuju između relacionog i XML formata. Ovi zahtevi su podržani specifikacijom SQL/XML, koja je uvedena kao Deo 14 (Part 14) standarda SQL:2003 i ažurirana 2006. SQL/XML uključuje novi XMLType tip podataka, skup funkcija za konverziju rezultata upita u XML format, i svojstvo kojim SQL može da pozove XQuery kao podjezik za obradu uskladištenih XML podataka.

Deo 14 SQL:2003 sadrži detaljnu definiciju novog XML tipa, vrednosti XML tipa, preslikavanje između SQL konstrukcija i XML konstrukcja i funkcije za generisanje XML podataka iz SQL podataka. Neke od funkcija za generisanje XML-a iz SQL podataka su:

XMLElement() – uzima ime elementa, opcionu kolekciju atributa za element i argumente koji čine sadržaj elemenat, i vraća instancu (vrednost) tipa XMLType

XMLForest() konvertuje svaki svoj parameter-argument u XML, i vraća XML fragment dobijen konkatenacijom tih konvertovanih argumenata

XMLConcat() uzima kao ulaz niz XMLType instanci, nadovezuje nizove elemenata u svakoj vrsti i vraća nadovezani niz (operator inverzan operatoru XMLSequence)

XMLAgg(), uzima kolekciju XML fragmenata i vraća agregirani XML dokument

XMLAttributes() – kreira listu atributa od polja SQL tabele

Na primer, iskaz

```
SELECT XMLELEMENT("Emp", XMLATTRIBUTES ( e.fname ||' '||
e.lname AS "name" ),  XMLForest ( e.hire, e.dept AS
"department")) AS "result" FROM employees e;
```

Porizvodi XML rezultat:

```
<Emp name="John Smith">
  <HIRE>2000-05-24</HIRE>
  <department>Accounting</department>
</Emp>
<Emp name="Mary Martin">
  <HIRE>1996-02-01</HIRE>
  <department>Shipping</department>
</Emp>
```

Standard SQL:2003 zamenio je prethodni standard, SQL:1999, i sa svim delovima ima više od 3.600 strana.

SQL:2006

SQL:2006 je samo dopuna standarda SQL:2003 koja se odnosi na podršku XML-u u bazama podataka. Preciznije, ISO/IEC 9075-14:2006 definiše kako se SQL koristi zajedno sa XML-om.

Definiše načine za importovanje i skladištenje XML podataka u SQL bazi podataka, kako se manipuliše njima u bazi podataka i kako se objavljuju (publikuju) XML podaci i konvencionalni SQL podaci u XML obliku.

SQL:2006 obezbeđuje mogućnost integracije (u aplikaciji) SQL koda i XQuery, XML upitnog jezika (W3C), kao i konkurentni pristup "običnim" SQL podacima i XML dokumentima.

SQL:2008

Uključuje izmene iz 2006 kao reviziju drugih delova SQL 2003 (na primer, MERGE iskaz).

SQL:2011

Revizija standarda iz 2008, npr. uključenjem DELETE operacije u MERGE iskaz, pipelined DML (uključenje iskaza ažuriranja u SELECT iskaz), itd.

Na primer, za tabele Inventory(Part, Qty), Changes(Part, Qty, Action), DELETE u MERGE:

```
MERGE INTO Inventory AS I
USING Changes AS C ON I.Part = C.Part
WHEN MATCHED AND   C.Action = 'Mod'   THEN
 UPDATE   SET Qty = Qty + C.Qty
WHEN MATCHED AND  C.Action = 'Dis'   THEN
 DELETE
WHEN NOT MATCHED AND  C.Action = 'Mod' THEN
INSERT   VALUES (C.Part, C.Qty)
```

Pipelined DML odnosi se na INSERT, UPDATE, DELETE u SELECT naredbi, označavajući skup koji se unosi (briše, ažurira) kao NEW TABLE (OLD TABLE):

Primer:

```
SELECT Oldtable.Empno
FROM OLD TABLE (DELETE FROM Emp WHERE Deptno = 2)
          AS Oldtable

SELECT Newtable.Empno
FROM NEW TABLE (UPDATE EMP
                          SET Salary = 0
                         WHERE Empno > 100)
         AS Newtable
```

SQL: 2016

Row Pattern Recognition
JSON
Polymorphic Table Functions

LITERATURA

http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt Information Technology - Database Language SQL

SQL:1999, formerly known as SQL3, Andrew Eisenberg, Jim Melton
SQL Bible http://www.tar.hu/sqlbible/sqlbible0014.html
SQL:2003 Has Been Published, Andrew Eisenberg, Jim Melton, Krishna Kulkarni, Jan-Eike Michels, Fred Zemke, SIGMOD Record, Vol. 33, No. 1, March 2004
SQL 2003 Standard Support in Oracle Database 10g , http www.oracle.com technolog
http://en.wikipedia.org/wiki/SQL
What's new in SQL:2011?
http://sigmod.org/publications/sigmodRecord/1203/pdfs/10.industry.zemke.pdf

https://standards.iso.org/ittf/PubliclyAvailableStandards/c067367_ISO_IEC_TR_19075-6_2017.zip

https://modern-sql.com/blog/2017-06/whats-new-in-sql-2016