

Objektno-relacijski sistemi za upravljanje podatkovnih baz - SQL3

Iztok Savnik

FAMNIT, 2011/12

Potek predstavitve

- Razvoj SQL standardov
- SQL2
- SQL3 - Pregled vsebine
- SQL3 - Pregled gradnikov
- Zaključki
- Literatura

Razvoj SQL Standardov

- Organizacije: ISO, ANSI

SQL/86	1987
SQL/89	1989
SQL2 (92)	1992
SQL3	1998/99
SQL/03	2003
SQL/07	2007

Večina delov standarda se je končalo v 2003 in 2007.

Potek standardizacije (1)

- 1970 Ted Codd - članek o relacijskem modelu
- 1975 CODASYL Database Specifications
- 1977 Database Project Initiated in U.S.
- 1978 ANSI Database Project Approved
- 1979 ISO Database Project Initiated
- 1982 ANSI Project Split into NDL and SQL
- 1983 ISO Project Split into NDL and SQL
- 1986 ANSI SQL Published – December
- 1987 ISO/IEC 9075:1986 (SQL86)
- 1989 ISO/IEC 9075:1989 (SQL89)
- 1992 ISO/IEC 9075:1992 (SQL92)
- 1995 ISO/IEC 9075-3:1995 (SQL/CLI for SQL92)
- 1996 ISO/IEC 9075-4:1996 (SQL/PSM for SQL92)

Potek standardizacije (2)

- SQL/86
- SQL/89 (FIPS 127-1)
- SQL/92
 - Vstopni nivo (FIPS 127–2)
 - Vmesni nivo I
 - Kompletan nivo
- SQL CLI
- SQL PSM
- SQL/3
 - SQL Framework
 - SQL Foundation
 - SQL Call Level Interface (CLI)
 - SQL Persistent Stored Modules (PSM)
 - SQL Language Bindings
 - SQL Management of External Data
 - SQL Object Language Bindings
- SQL/4
 - All of the above, and ...
 - XA
 - SQL Temporal

SQL/86 (ISO/IEC 9075:1986)

- Začetna točka: IBM SQL implementacija
- SQL/86 postane podmnožica IBM SQL implementacije
- Trije načini procesiranja DML
 - Direktno procesiranje
 - Jezik modulov
 - Vgnezden SQL
- Povezave z:
 - Cobol
 - Fortran
 - Pascal
 - PL/1

SQL/89 (ISO/IEC 9075:1989)

- Nad-množica SQL/86
- Zamenja SQL/86
- C in ADA sta dodana k obstoječim povezanim jezikom
- DDL v posebnem jeziku „jezik za definicijo sheme”
 - CREATE TABLE, CREATE VIEW, GRANT PRIVILEGES (No DROP, ALTER, REVOKE),
 - DEFAULT, UNIQUE, NOT NULL, Views WITH CHECK OPTION, PRIMARY KEY. CHECK constraint
- Referenčna integriteta

SQL/92 (1)

- Nad-množica SQL/89
- Precej večji od SQL/89 (579 vs 120 strani)
 - Razširitve podatkovnih tipov (varchar, bit, char sets, date, time&interval)
 - Več operacij stikov
 - Katalogi
 - Domene
 - Izpeljane tabele v FROM stavku
 - Dejstva (assertions)
 - Začasne tabele
 - Akcije referenčne integritete
 - Jezik za delo s shemo
 - Dinamični SQL, pregledovanje kursorjev,
 - Povezave (connections)

SQL/92 (2)

- Veliko (ne vsi) gradnikov iz obstoječih DBMS
- Razdeljen na 3 nivoje
 - Vstop nivo (SQL/89 z izboljšanjem integritetnih omejitev)
 - Vmesni nivo
 - Celoten nivo
- Gradniki so prirejeni nivojem
 - Celoten nivo vsebuje vmesnega
 - Vmesni vsebuje vstopnega
- FIPS 127-2 definira še Prehodni nivo
 - Nivo med vstopnim in vmesnim
 - Podmnožica vmesnega
 - Nadmnožica vstopnega

SQL/92 (3)

- SQL/92 po nivojih
- Vstopni nivo
 - SQL/89 + nekateri gradniki
- Vmesni nivo
 - Podmnožica SQL/92
 - Vstopni nivo + Kursorji, domene, DDL, implicitne pretvorbe tipov, operacije nad bloki poizvedb, CASE, UNIQUE, več-jezične kodne tabele, bogatejši katalog (omejitve, domene, ...), ...
- Kompletan nivo
 - Vmesni + izpeljane tabele, referenčna integriteta z akcijami, integritetna dejstva, razširjeni predikati WHERE, ve tipov stikov, več osnovnih tipov, začasne tabele, eksplicitne tabele, DISTINCT na pod-izrazih, popravki preko kursorjev, UPDATE in DELETE preko pod-vprašanj na isti tabeli, ...

SQL2 (SQL/92)

- začetek dela 1989
- identična ISO in ANSI standarda
- pokriva SQL/86 in SQL/89
- nivoji
 - osnovni, vmesni, celotni
- objava konec 1992
- trenutno uporabljan standard

Pregled SQL2

- Tabele
- Integritetne omejitve
- Povpraševalni izrazi
- Okna
- Varnost
- Dinamični SQL

SQL2 - Tabele

- osnovni podatkovni tipi:
 - CHAR(), VARYING(), NUMERIC(), ...
- domene:
 - CREATE DOMAIN
- tabele:
 - CREATE TABLE
 - ALTER TABLE
 - DROP TABLE

SQL2 - Povpraševalni izrazi

- stavek SELECT
SELECT kraj FROM zaposleni
WERE starost=32;
- stavek WHERE (pogoji)
 - NOT, IN, ALL, ANY, SOME, EXISTS, ...
- dodatni stavki:
 - HAVING, GROUP BY, ORDER BY

SQL2 - Povpraševalni izrazi (2)

- operacije:
 - UNION, EXCEPT, INTERSECT, JOIN, RIGHT JOIN, LEFT JOIN, OUTER JOIN, ...
- stavki:
 - INSERT, UPDATE, DELETE, ALTER

SQL2 - Okna

- virtualne tabele definirane s **SELECT**
CREATE VIEW dbskupina **AS**
SELECT * FROM zaposleni
WHERE delo='dba';
- operacije **UPDATE**, **INSERT** in **DELETE** delujejo tudi nad okni
- določilo: **WITH CHECK OPTION**

SQL2 - Integritetne omejitve

- omejitve tabel in stolpcev:
 - PRIMARY KEY, UNIQUE, FOREIGN KEY, .
- akcije referenčnih omejitev
 - ON DELETE (UPDATE) ==> NO ACTION, CASCADE, SET DEFAULT, SET NULL
- kompleksne omejitve:
 - CHECK(p) - 'p' lahko vsebuje poljubni povpraševalni izraz
 - ASSERTION

SQL2 - Varnost

- avtorizacijski identifikator
 - identifikator sej, modulov, uporabnika, ...
 - določilo AUTHORIZATION
- privilegiji in pravila dostopa
 - GRANT
 - REVOKE

SQL2 - Dinamični SQL

- dinamični SQL: stavek
 - SQL stavek vsebujeje spremenljivke
 - struktura vhodnih tabel ali rezultata ni znana
- kursorji
 - DECLARE, OPEN, FETCH, UPDATE, ...
- dva načina izvajanja dinam. SQL stavkov
 1. EXECUTE IMMEDIATE
 2. PREPARE + EXECUTE ... USING

SQL2 kompatibilni produkti

- validacija
 - NIST: National Institute of Standards and Technology
- SQL2
 - osnovni nivo: 10 sistemov na 100 platformah
 - višji nivoji: testno okolje definirano leta 1996

SQL3 - Pregled vsebine

- Deli standarda
 - Framework, Foundtions, CLI, PSM, ...
- Predstavitev delov standarda
 - Pregled osnovnih lastnosti
- Trenutni status standarda
 - Verzije, zaključeni deli, predlogi standarda

Deli SQL3

- SQL/Framework (Z-03)
- SQL/Foundations (Z-03)
- SQL/CLI (Z-03)
- SQL/PSM (-> Found)
- SQL/Bindings (-> Found)
- SQL/XA (Cancel)
- SQL/Temporal (Wdrw)
- SQL/Object (-> Found)
- SQL/OLB (Z-03)
- SQL/MED (Z-03)
- SQL/JRT (Z-03)
- SQL/XML (not cml)

SQL/Framework

- Pregled celotnega standarda
- Definira kako so deli povezani med sabo
- Vsebuje skupne specifikacije
 - definicije, koncepti, konvencije

SQL/Foundations

- Novi osnovni tipi
 - boolean
 - large object
- Sistem tipov
 - ADT
 - kompleksni objekti
 - podtipi, dedovanje
 - enkapsulacija
 - funkcije, ...
- Shranjeni podprogrami
 - prekrivanje
 - statično preverjanje tipov
- Rekurzija
- Prožilci (triggers)
- Vloge (roles)

SQL/Foundations (1)

- Celotna funkcionalnost SQL/92
 - Scheme, različni stiki, začasne tabele, CASE izrazi, cursorji, ...
- Novi vgrajeni tipi
 - Boolean, Large objects (LOBs)
- Izboljšano delo update stavkov
 - Update/delete skozi unije/stike
- Druge relacijske razširitve
 - Dodatni predikati (FOR ALL, FOR SOME, SIMILAR TO)
 - Razširitve cursorjev
 - Razširitve referenčne integritete (RESTRICT)
 - Razširitve stikov

•

SQL/Foundations (2)

- Prožilci: izboljšana integritetni mehanizmi
 - Različni prožilni dogodki: update/delete/insert
 - Opcijski pogoji
 - Čas aktivacije: pred / potem
 - Akcija z večimi stavki
 - Več prožilcev na eno tabelo
 - Pogoj in akcija (več stavkov) za vsako vrstico ali vsak stavek
- Vloge (role): Izbolšan sistem varnosti
 - GRANT/REVOKE privilegiji vlogam
 - GRANT/REVOKE vloge uporabnikom in drugim vlogam

SQL/Foundations (3)

- Rekurzija
 - Povečana izrazna moč
 - Linearna (direktna in izključujoča) rekurzija
 - Ustavitveni pogoji
 - Različne iskalne strategije (v globino, v širino)
- Kontrolne točke
 - Izboljšana uporabniška kontrola nad integriteto
 - Definicija shranjevalnih točk
 - Vrnitev na (Roll back) na shranjevalno točko
 - Gnezdenje
- OLAP razširitve:
 - izboljša zmožnost povpraševalnega jezika
 - CUBE, ROLLUP, Izrazi v ORDER BY

SQL/Foundations (4)

- Objektno-relacijske razširitve
 - Razširljivost
 - Povečana izrazna moč (kompleksni objekti)
 - Ponovna uporabnost
 - Integracija
- Uporabniško definirani tipi
 - Enostavni tipi
 - Strukturirani tipi
 - Močni tipi, obnašanje odvisno od tipa, enkapsulacija, zamenljivost vrednosti, polimorfizem, dinamično povezovanje, statični tipi

SQL/Foundations (5)

- Kolekcije
 - Polja, sezname, množice, multi-množice
- Konstruktor ROW
 - Kot zapisi v programskih jezikih
 - Tip vrstic v tabelah
 - Gnezdenje (vrstice vsebujejo polja, ki so vrstice)
- Tipi referenc
 - Omogočijo “objektni identifikator”
 - Navigacijski dostop (izrazi poti)

SQL/Foundations (6)

- Uporabniško-definirane funkcije
 - SQL in zunanje funkcije, prekrite funkcije, uporabniško-definirane poti, statično preverjanje tipov, statično povezovanje
- Uporabniško-definirane procedure
 - SQL in zunanje funkcije, ni prekrivanja, vhodni in izhodni parametri, rezultatne množice, statično povezovanje
- Uporabniško-definirane metode
 - Predstavljajo uporabniško definirano obnašanje, SQL in zunanje metode, preobloženost in prekrivanje, dinamični preverjanje tipov, zakasnjeno povezovanje

SQL/Foundations (7)

- Hierarhije tabel
 - Povečana izrazna zmožnost modeliranja in vprašanj
 - Orodje za modeliranje specializacije /generalizacije
 - CREATE/DROP subtable
 - CREATE/DROP subview
 - Objektni “identifikator” kot referenca
 - Vprašanja delujejo tudi nad pod-tabelami
 - “Objektni stil” dela preko referenc in izrazov poti
- Hierarhije pogledov (hierarhija objektov)

SQL/CLI

- Call Level Interface (CLI)
- Omogoča izvajanje SQL stavkov iz aplikacij preko standardnega vmesnika
 - deluje podobno kot Vgnezdni SQL
 - osnovan na ODBC, SQL Access Group (SAG)
 - omogočen dostop do pomembnejših podatkovnih struktur strežnika
- Ni potrebna uporaba predprevajalnika

SQL/PSM

- shranjeni podprogrami in funkcije
- razširja izrazno moč SQL
- lastnosti podprogramov in funkcij
 - prekrivanje, resolucija po vseh argumentih
 - statično preverjanje tipov
- gradnik programskih jezikov
 - case, if, loop, for, begin/end block
 - delo z izjemami, itd.

SQL/Bindings

- Sintaksa SQL stavkov vgnezenih v programske jezike
- Primerjava z SQL2
 - Dinamični SQL iz SQL2
 - Razširitve standardnih podatkovnih struktur
 - SQLSTATE, SQLEXCEPTION, ...
 - Razširitve modulov klienta
 - Razširitve pravic za delo s podprogrami

SQL/XA

- Gradniki za realizacijo upravljalnika transakcij, ki uporablja več strežnikov
- Vmesnik XA
 - definira komunikacijo med moduli upravljalnika transakcij
 - X/Open; sprejet kot standard
- Porazdeljeni sistemi za upravljanje podatkovnih baz

SQL/Temporal

- Razširitve SQL z gradniki za delo s podatki vezanimi na čas
- Nov podatkovni tip
 - PERIOD
- Operacije in predikati za delo s časovnimi intervali
 - normalizacija, pred, za, prekrivanje, vsebovanost, itd.

SQL/Object

- Razširitve sistema tipov z objekti
- Združitev relacijskih in OO konceptov
- Povečana izrazna moč jezika
 - uporabljen model kompleksnih objektov
 - povečan obseg ciljnih aplikacij
- Možnost ponovne uporabe

SQL/OLB

- Java & SQL
 - razširitve SQL, ki omogočajo njegovo uporabo v Javi
- Primerjava z Vgnezdenim SQL
- Definirani so Javini moduli in razredi za delo z SQL

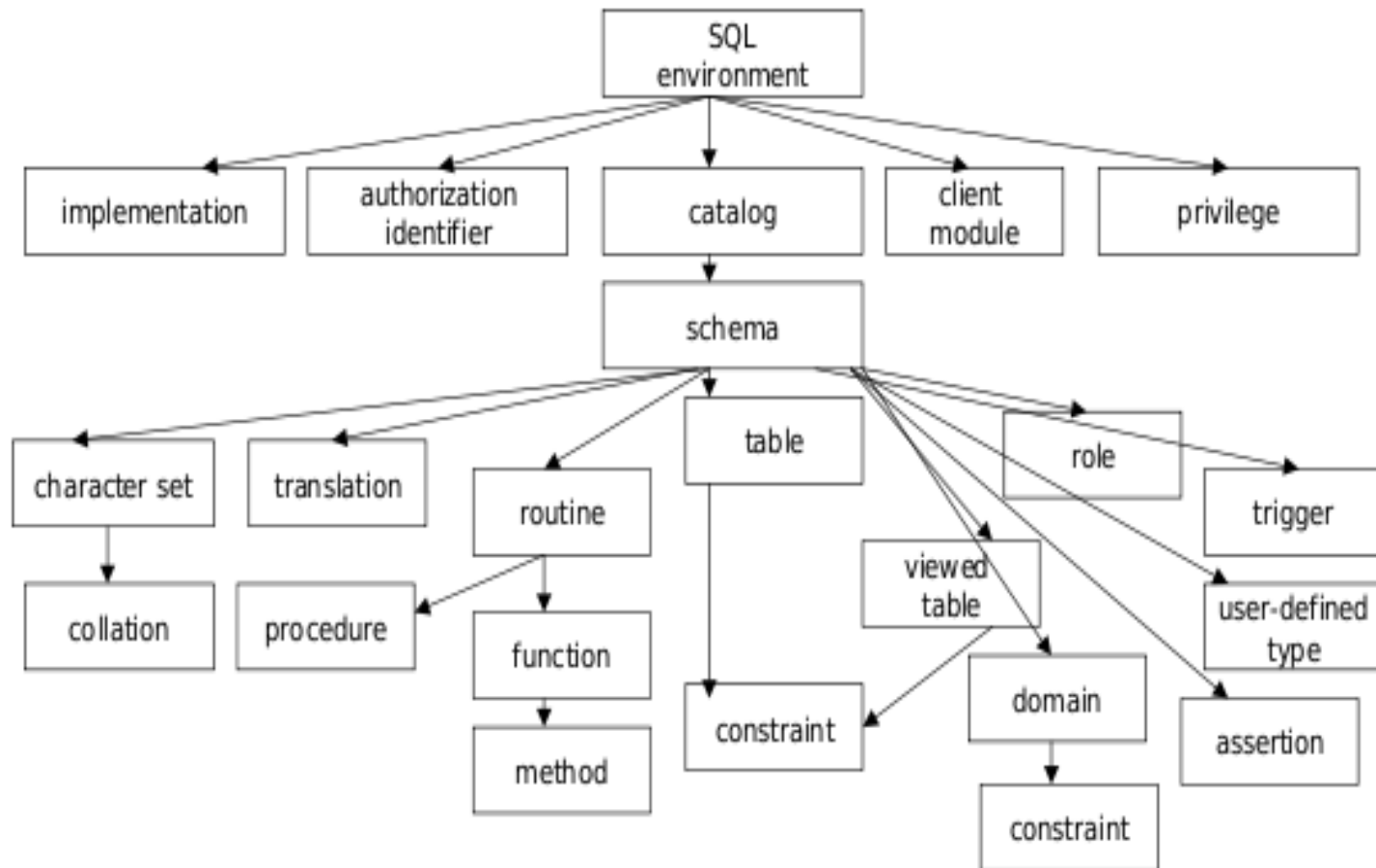
SQL/MED

- Razširitev SQL za delo z eksternimi podatki
 - HTML, datoteke, slike, itn.
- Abstraktne tabele
- Abstraktni veliki objekti (LOB)
- Podatkovne povezave (datalink)

SQL3 - Pregled gradnikov

- Podatkovni tipi
- Abstraktni podatkovni tipi
- Specifične poizvedbe
- Rekurzija
- Prožilci

Objekti PB



Katalogi in sheme

- SQL objekti (npr. tabele, pogledi, ...) so sestavni del sheme
- Sheme so sestavljajo kataloge
- Vsaka shema ima enega samega lastnika
- Objekte lahko referenciramo z eksplicitnim ali implicitnim katalogom in shemo

FROM people

FROM sample.people

FROM cat1.sample.people

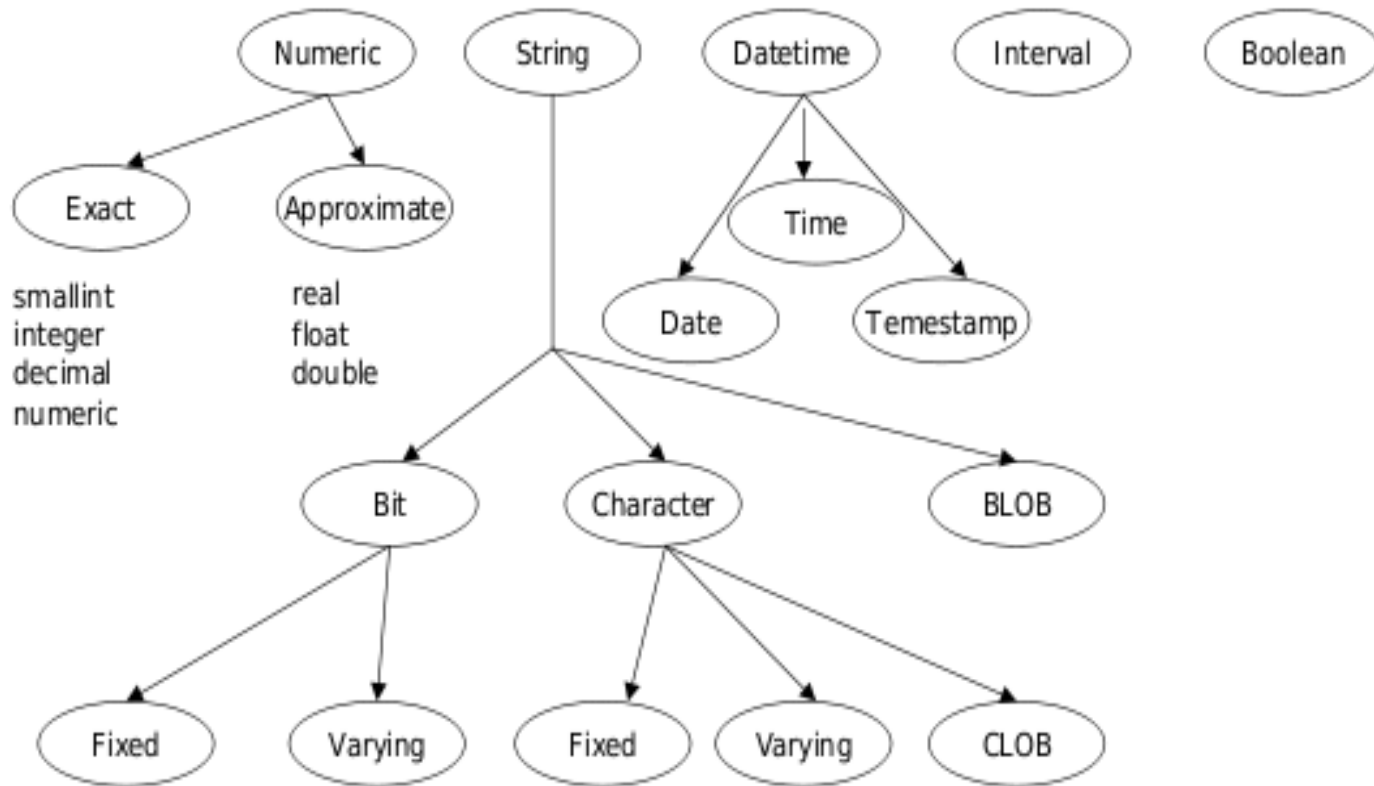
Jezik za delo s shemo

- Kreiranje, brisanje, spreminjanje objektov sheme
- Brisanje ima dva obnašanja
 - RESTRICT, CASCADE
- Table
 - Add/drop column, Alter column default, Add/drop constraints
- Domain
 - Set/drop default, Add/drop omejitev
- Uporabniško definirani tipi
 - Add/drop atribut, Add/drop metoda
- SQL rutine
 - Alter routine - lastnosti

Podatkovni tipi

- Predefinirani tipi
 - Numeric, String, BLOB, Boolean, Datetime, Interval
- Konstruirani atomični tipi
 - Reference
- Konstruirani sestavljeni tipi
 - Kolekcija: Array
 - Row
- Uporabniško definirani tipi
 - Distinct tip
 - Strukturiran tip

Predefinirani tipi



Domene

- Persistentna, imenovana definicija:
 - Podatkovnega tipa, opsijska privzeta vrednost, opsijska množica omejitev, opsijska sekvenca
 - Uporabljena namesto podatkovnega tipa

```
CREATE DOMAIN money AS DECIMAL (7,2);
CREATE DOMAIN account_type AS CHAR (1)
DEFAULT 'C'
CONSTRAINT account_type_check CHECK (value IN ('C','S','M'));
CREATE TABLE accounts
(account_id INTEGER, balance money, type account_type);
```

SQL Rutine

- Imenovana persistentna koda, ki jo aktiviramo iz SQL
 - SQL procedure, funkcije in metode
- Kreirane v shemi ali v strežniškem modulu SQL
 - Rutine na nivoju sheme in modulov
- Podprt DDL
 - CREATE in DROP stavki
 - ALTER stavki – omejeni v funkcionalnosti
 - EXECUTE privilegij, ki se kontrolira preko GRANT in REVOKE stavka

SQL Rutine (1)

- Rutine imajo glavo in telo
- Glava vsebuje ime in (lahko prazen) seznam parametrov
 - Parametri so lahko: IN, OUT, INOUT
 - Parametri funkcij so vedno IN
 - Funkcije vrnejo eno vrednost določeno z RETURNS stavkom
- SQL routine
 - Glava in telo sta zapisana v SQL
- Eksterne rutine
 - Glava je opisana v SQL
 - Telo je zapisano v gostiteljskem jeziku
 - Lahko vsebuje vgnezden SQL

SQL Rutine (2)

- Parametri morajo imeti ime in tip
- Telo rutine sestavljeno iz enega SQL stavka
 - Lahko vsebuje : BEGIN ...END
 - Ne sme vsebovati: DDL, CONNECT, DISCONNECT, dinamični SQL, COMMIT, ROLLBACK

```
CREATE PROCEDURE get_balance(IN acct_id INT, OUT bal
DECIMAL(15,2))
BEGIN
    SELECT balance INTO bal
        FROM accounts WHERE account_id =acct_id;
    IF bal <100
        THEN SIGNAL low_balance
    END IF;
END
```

SQL Rutine (3)

- Telo rutine
 - RETURN stavek je dovoljen samo v funkciji
 - Izjema se sproži, če se funkcija zaključi brez RETURN

```
CREATE FUNCTION get_balance(acct_id INT) RETURNS
DECIMAL(15,2)
BEGIN
    DECLARE bal DECIMAL(15,2);
    SELECT balance INTO bal
        FROM accounts
        WHERE account_id = acct_id;
    IF bal <100 THEN SIGNAL low_balance
    END IF;
    RETURN bal;
END
```

SQL Rutine (4)

- Parametri
 - Imena so opcijška
 - Ni dovoljen katerikoli SQL tip (odvisno od gostitelja)
- LANGUAGE: Identificira gostiteljski jezik
- NAME: Identificira datoteko s kodo

```
CREATE PROCEDURE get_balance (IN acct_id INT, OUT bal
DECIMAL(15,2))
LANGUAGE C
EXTERNAL NAME 'bankSbalance_proc'
```

```
CREATE FUNCTION get_balance (IN INTEGER) RETURNS
DECIMAL(15,2)
LANGUAGE C
EXTERNAL NAME 'usr/han/banking/balance'
```

SQL Rutine: večobličnost

- Imamo več rutin z istim imenom
 - S1.F (p1 INT, p2 REAL)
 - S1.F (p1 REAL, p2 INT)
 - S2.F (p1 INT, p2 REAL)
- Znotraj iste sheme moramo imeti enolično signaturo
 - S1.F (p1 INT, p2 REAL)
 - S1.F (p1 REAL, p2 INT)
- Med shemami lahko imamo isto signaturo
 - S1.F (p1 INT, p2 REAL)
 - S2.F (p1 INT, p2 REAL)
- Prekrite so lahko samo funkcije

SQL Rutine: klic

- Proceduro sprožimo s stavkom CALL
 - `CALL get_balance(100, bal);`
- Funkcija je lahko poklicana znotraj izraza
 - `SELECT account_id, get_balance (account_id)`
 - `FROM accounts`
- Zahteva, da imamo EXECUTE privilegij za klicano rutino
 - Sicer ne najde rutine

Objektno-relacijski model

- Razširitev osnovnih tipov
 - BOOLEAN, LOB-s
- Strukturirani tipi
 - ROW, TABLE
- Imenovani tipi
 - ADT, Distinct Type
- Konstruktorji tipov
 - REF, SET, LIST, MULTISSET

LOB

- LOB-i so novi podatkovni tip
- LOB-i shranjujejo nize dolge več GB
- Dva nova tipa:
 - BLOB - Binary Large Object
 - Audio, slike, video
 - CLOB - Character Large Object
 - Tekst

LOB (1)

- Shranjeni v podatkovni bazi
 - Ne v “zunanjih datotekah”
- Velikost LOB se določi ob kreaciji zapisa
 - (KB, MB, GB)

```
CREATE TABLE BookTable
(title          varchar(200),
book_id        INTEGER,
summary        CLOB(32K),
movie          BLOB(2G));
```


LOB: Zgled

```
CREATE TABLE zaposleni
( zap#    INTEGER,
  ime     VARCHAR(30),
  ...
  povzetek CLOB(75K),
  podpis  BLOB(1M),
  slika   BLOB(10M),
  redno   BOOLEAN);
```

LOB: Popravljanje

- LOB-e lahko preberemo, vstavimo, popravimo kot vse ostale tipe
- Imeti je potrebno zadosti velike vmesnike za LOB
 - Težko za zelo velike LOB
 - SQL99: lokatorji

```
EXEC SQL
```

```
  SELECT summary, book_text, movie  
  INTO :bigbuf, :biggerbuf, :massivebuf  
  FROM BookTable  
  WHERE title="Moby Dick";
```

LOB: Funkcije

- Funkcije LOB
 - CONCATENATION string1|| string2
 - SUBSTRING(string FROM start FOR length)
 - LENGTH(expression)
 - POSITION(search-string IN source-string)
 - NULLIF/COALESCE
 - TRIM, OVERLAY, Cast
 - Uporabniško-definirane funkcije
 - LIKE

```
EXEC SQL
    SELECT position(„Chapter 1“ IN book_text
    INTO :int_variable
    FROM BookTable
    WHERE title=„Moby Dick“;
```

Uporabniško definirani tipi

- Uporabniško definirani, imenovani tipi
 - zaposleni, projekt, denar, poligon, slika, tekst, jezik, ...
 - (1) Razlikovalni tipi
 - Osnovani na predefiniranih tipih
 - Ni dedovanja
 - (2) Strukturirani tipi
 - Eden ali več atributov
 - Podprta je hierarhija tipov
- Uporabniško definirane metode in funkcije
 - Uporabniško definirane operacije predstavljajo obnašanje uporabniških objektov v aplikacijski domeni
 - hire, appraisal, convert, area, length, contains, ranking, ...

•

Razlikovalni tipi

```
CREATE TYPE plan.roomtype  
AS CHAR(10);
```

```
CREATE TYPE plan.meters  
AS INTEGER;
```

```
CREATE TYPE plan.squaremeters  
AS INTEGER;
```

```
CREATE TABLE RoomTable  
(RoomID      plan.roomtype,  
 RoomLength  plan.meters,  
 RoomWidth   plan.meters,  
 RoomPerimeter plan.meters,  
 RoomArea    plan.squaremeters);
```

```
UPDATE RoomTable  
SET RoomArea=RoomLength;
```

Napaka v tipu !!!

Strukturirani tipi: ADT

- Abstraktni podatkovni tipi
 - Imenovani tip, ki ga definira uporabnik
 - Opis obnašanja in interne strukture
- Kompleksne strukture
- Funkcije
- Podtipi in dedovanje
- Enkapsulacija

Strukturirani tipi: ADT (1)

- “Od zunaj” ni razlike med atributi, funkcijami in virtualnimi atributi
- Fizična predstavitev se lahko spreminja brez vpliva na zunanjo aplikacijo
- Javni/privatni atributi in funkcije
- Kreiranje primerkov s konstruktorjem

Strukturirani tipi: ADT (2)

- Tip stolpca
 - Text, slika, audio, video, časovne vrste, točka, premica, ...
- Konstruktorji tipov
 - REF
 - ROW, TABLE
 - LIST, SET, MULTISSET
- Ortogonalnost konstruktorjev
- Povpraševanja nad kompleksnimi strukturami

Strukturirani tipi: Primer

```
CREATE TYPE address AS
(street    CHAR(30),
 city     CHAR(20),
 state    CHAR(2),
 zip      INTEGER);
```

```
CREATE TYPE bitmap AS BLOB;
```

```
CREATE TYPE real_estate AS
(owner          REF(person),
 price         money,
 rooms        INTEGER,
 size         DECIMAL(8,2),
 location     address,
 text_description text,
 front_view_image bitmap,
 Document     doc);
```

Strukturirani tipi: Podtipi (1)

- Podtip je vsebovan v nadtipu
- `CREATE TYPE` vira `UNDER` hiša ...;
- En tip ima lahko več nadtipov
- Princip zamenljivosti:
 - primerki podtipov se lahko uporabijo kjerkoli se lahko uporabijo primerki nadtipov
- Zamenjava tipa primerka

Strukturirani tipi: Podtipi (2)

- Podtip podeduje vse lastnosti nadtipov
 - Atributi nadtipa se dedujejo na podtipe
 - Primerek podtipa lahko uporablja funkcije nadtipov
- Izbira funkcije
 - Prekrivanje, večkratno dedovanje
 - Glede na parametre funkcije

Metode

- Kaj so metode?
 - SQL funkcije pripete na uporabniško definirane tipe
- Kako so drugačne od funkcij?
 - Implicitna spremenljivka SELF
 - Predstavljena s signaturo in telesom metode
 - Mora biti kreirana v shemi tipa
 - Različni načini klica

```
CREATE TYPE employee AS
(name          CHAR(40),
 base_salary   DECIMAL(9,2),
 Bonus         DECIMAL(9,2),
 INSTANTIABLE NOT FINAL
 METHOD salary() RETURNS DECIMAL(9,2));
```

```
CREATE METHOD salary() FOR employee
BEGIN
...
END;
```

Metode (1)

- Originalne metode: metode super-tipa
- Prekrivajoče metode: metode sub-tipa

```
CREATE TYPE manager UNDER employee
(stock_option INTEGER,
  OVERRIDING METHOD salary() RETURNS DECIMAL(9,2),
  METHOD vested() RETURNS INTEGER);
```

- Klicanje metod s sintakso z uporabo pike

```
SELECT mgr.salary() FROM dept;
```

Kreiranje strukturiranih tipov

- Sistemska funkcija: konstruktor
 - `address()` -> `address` `real_estate()` -> `real_estate`
 - Vrne objekt z inicializiranimi atributi
- Operator `NEW`
 - `NEW <method name> <list of parameters>`
- Poglejmo si primer `INSERT` stavka

```
CREATE TABLE properties OF real_estate ...
```

```
INSERT INTO properties VALUES (:owner, money(350000), 15, 4500,  
NEW address('1543 North', ...
```

Strukturirani tipi: Primer

```
CREATE TABLE zaposleni
( zap#      INTEGER,
  ...
  naslov    ROW( ulica   VARCHAR(40),
                 mesto   VARCHAR(20),
                 hštev   INTEGER,
                 država   VARCHAR(30)),
  otroci    LIST (oseba),
  hobiji    SET (VARCHAR(20)));
```

Podtipi in dedovanje

- Strukturirani tipi so lahko podtipi drugih ADT
- ADT podeduje strukturo in obnašanje od nad-tipov
- FINAL in NOT FINAL

```
CREATE TYPE real_estate ... NOT FINAL
```

```
CREATE TYPE condo UNDER real_estate ... NOT FINAL
```

```
CREATE TYPE house UNDER real_estate ... NOT FINAL
```


Podtipi in dedovanje (1)

```
CREATE TYPE address AS
```

```
(street CHAR(30), city CHAR(20), state CHAR(2), zip INTEGER)  
NOT FINAL
```

```
CREATE TYPE german_addr UNDER address
```

```
(family_name VARCHAR(30)) NOT FINAL
```

```
CREATE TYPE brazilian_addr UNDER address
```

```
(neighborhood VARCHAR(30)) NOT FINAL
```

```
CREATE TYPE us_addr UNDER address
```

```
(area_code INTEGER, phone INTEGER) NOT FINAL
```

```
CREATE TYPE us_bus_addr UNDER us_addr
```

```
(bus_area_code INTEGER, bus_phone INTEGER) NOT FINAL
```

Strukturirani tipi: Tabele

- Strukturirane tipe uporabimo za definicijo tabel
 - Atributi tipa postanejo stolpci tabele
 - Plusen stolpec, ki definira REF vrednost

```
CREATE TYPE real_estate AS
(owner          REF (person),
 Price         money,
 Rooms        INTEGER,
 Size         DECIMAL(8,2),
 Location     address,
 text_description      text,
 front_view_image     bitmap,
 Document            doc) NOT FINAL
```

```
CREATE TABLE properties OF real_estate
(REF IS oid USER GENERATED)
```

Strukturirani tipi: Reference

- Strukturirani tipi imajo pripadajoč referenčni tip
 - Lahko ga uporabljamo kjerkoli lahko uporabimo druge tipe
- Predstavitev
 - Uporabniško generiran (REF USING <predefined type>, REF FROM (<attr-names>), REF IS <self-ident attr>)
 - Sistemsko generiran (REF IS SYSTEM GENERATED) : default
 - Definiran kot seznam atributov (REF (<list of attributes>))

```
CREATE TYPE real_estate AS (owner REF (person), ...)
NOT FINAL REF USING INTEGER
```

```
CREATE TYPE person AS (ssn INTEGER, name CHAR(30), ...)
NOT FINAL REF (ssn)
```

Strukturirani tipi: Hierarhije tabel

- Tabele imajo lahko podtabele
 - Podedujejo stolpce, omejitve, prožilce, ... supertabel

```
CREATE TYPE person... NOT FINAL
CREATE TYPE real_estate ... NOT FINAL
CREATE TYPE  condo UNDER real_estate ... NOT FINAL
CREATE TYPE  house UNDER real_estate ... NOT FINAL
```

```
CREATE TABLE people OF person (...)
CREATE TABLE properties OF real_estate
CREATE TABLE condos OF condo UNDER properties
CREATE TABLE houses OF house UNDER properties
```

SQL3 - specifične poizvedbe

- dostop do komponent zapisa
 - uporaba “poti”
- povpraševanje nad vgnezdenimi množicami
 - selekcija, gnezdenje, razgnezdenje
- delo z referencami
 - vrednost reference
 - posebni stolpec tabele

Zgled: vgnezdene strukture

- Poišči podatke o zaposlenih, ki radi potujejo in imajo več kot pet otrok.

```
SELECT *  
FROM zaposleni z  
WHERE 'potovanja' IN  
    (SELECT *  
     FROM TABLE(z.hobiji) h)  
AND 5 < (SELECT count(*)  
        FROM TABLE(z.otroci) o);
```

Zgled: razgnezdenje

- Poisci otroke vseh zaposlenih, ki zaslužijo manj kot 70000.

```
SELECT o.*  
FROM zaposleni z, TABLE (z.otroci) o  
WHERE z.placa < 70000;
```

Zgled: kreiranje struktur

- Poišči šefe oddelkov in vse podatke o njihovih projektih, ki imajo finančno stanje > 10000.

```
SELECT o.šef,  
       MULTISET( SELECT *  
                 FROM TABLE(o.projekti) AS p  
                 WHERE p.fin_stanje > 10000 )  
FROM oddelki o;
```


SQL3 - Rekurzija

- Rekurzivne strukture
 - drevesa ali usmerjeni grafi
 - komponente zapisov so reference na zapise istega tipa
- Uporaba rekurzije
 - vprašanje: “določi ceno komponent sklopa”
 - preiskovanje mrež
- Cikli, duplikati, vrstni redi izpisa

Zgled: rekurzija

- Dana je binarna relacija 'Let', ki definira polete letal med mesti. Poišči vsa mesta b, ki so dosegljiva iz Ljubljane.

```
WITH RECURSION Dosegljiva_iz (a, b) AS
```

```
(SELECT a, b FROM Let
```

```
UNION
```

```
SELECT in.a, out.b
```

```
FROM Dosegljiva_iz in, Let out
```

```
WHERE in.b=out.a);
```

```
SELECT * FROM Dosegljiva_iz WHERE a='Ljubljana';
```

SQL3 - Prožilci

- Imenovana funkcija, ki se aktivira z definiranim dogodkom
- Aktivirana funkcija se izvrši, če je zadoščeno pogoju prožilca
- Uporaba:
 - validacija, izračunani atributi, poslovna pravila, enostaven razvoj in vzdrževanje aplikacije

Zgled: prožilec

- Definiraj prožilec, ki popravi stanje računa pred vstavljanjem zapisa v tabelo 'transakcije'.

```
CREATE TRIGGER popravi_stanje
BEFORE INSERT ON transakcije
  REFERENCING NEW AS tran
FOR EACH ROW
  WHEN (tran.TRAN_type='W')
  UPDATE računi
    SET stanje = stanje + tran.znesek
  WHERE račun#=tran.račun#;
```

Nekatere lastnosti prožilcev

- Aktivacijski dogodki: INSERT, UPDATE, ..
- Čas aktivacije: BEFORE, AFTER
- Prožilec lahko aktivira druge prožilce
- ES problem:
 - vrstni red aktiviranja prožilcev !
 - niso primerni za razvoj aplikacije

Postopkovni gradniki SQL3

- **Compound statement**
- **SQL variable declaration**
- **If statement**
- **Case statement**
- **Loop statement**
- **While statement**
- **Repeat statement**
- **For statement**
- **Leave statement**
- **Return statement**
- **Call statement**
- **Assignment statement**
- **Signal/ resignal statement**
- **BEGIN ... END;**
- **DECLARE var CHAR(6);**
- **If subject (var<>'urgent')**
THEN ... ELSE ...;
- **Case subject (var)**
WHEN 'SQL' THEN ...
WHEN ...;
- **Loop <SQL statement list>**
END LOOP;
- **While i<100 DO ... END WHILE;**
- **REPEAT ... UNTIL i<100 END**
REPEAT;
- **For result AS ... DO ... END**
FOR;
- **LEAVE ...;**
- **RETURN 'urgent';**
- **CALL procedure_x (1,3,5);**
- **SET x='abc';**
- **SIGNAL division_by_zero**

Zaključki

- SQL3 je navzgor kompatibilen z SQL2
 - razširitev relacijskega jedra
- Objektno-relacijski podatkovni model
 - vsebuje vse dobre lastnosti OO modela
- Vsebuje gradnike Aktivnih DBMS
 - uporaba prožilcev
- Razširjen nabor možnih aplikacij
 - CAD, procesni sistemi, multimedia, itd.

Zaključki (2)

- Razširljiv sistem tipov
- Računska kompletnost
- Možnost ponovne uporabe kode
- Možnost dela s tekstom in slikami
- Obsežnost standarda
- Kompleksnost

Literatura

Information technology, Database languages, SQL
ISO/IEC JTC 1/SC 32, ANSI, 2007.

A guide to the SQL standard

C.J.Date, H.Darwen, Addison-Wesley, 1994.

SQL Standards Home Page

http://www.jcc.com/sql_stnd.html

SQL Standards Archive Home Page

<ftp://jerry.ece.umassd.edu/isowg3>

SQL3 Standard Suite - An Overview

P.Pistor,DEXA, 1996.

Literatura (2)

An overview of SQL3 standard (folies)

N.Mattos, 1996.

Recent Design Trade-offs in SQL3

N.Mattos, L.DeMichiel, SIGMOD Record, 1996.

Influencing Database Language Standard

L.Gallagher, Članek, 1994.

SQL3 Standardization

Ki-Joon Han