

SQL

Богдан Шишеджиев - SQL

1

Версии

- SQL-92 е разработен на базата на SEQUEL (IBM) и в момента има два публикувани стандарта:
 - ANSI X3.135-1992, "Database Language SQL"
 - ISO/IEC 9075:1992, "Database Language SQL"
- Нива
 - Entry
 - Transitional
 - Intermediate
 - Full
- Всяка реализация трябва да включва "Entry" нивото

Богдан Шишеджиев - SQL

2

Подмножества на езика

- Език за дефиниция на данните – DDL
 - Дефиниране на домени
 - Дефиниране и промяна на схемата
 - Дефиниране на ограничения
 - Дефиниране на изгледи
 - Дефиниране на права за достъп
- Език за манипулация на данните – DML
 - Реализация на заявки (Query)
 - Добавяне на кортежи
 - Премахване на кортежи
 - Модификация на кортежи

Богдан Шишеджиев - SQL

3

Дефиниция на домена

- Елементарни домени
 - **Символни типове**
 - **Char** за отделни символи.
 - **Char(n)** за низове от *n* символа (**Varchar** при DB2)
 - **Varchar** (при ORACLE) за низове за връзка с процедурните езици.
 - **Long** (ORACLE) за низове от максимум 65655 символа.
 - В ACCESS те са Text, Memo.
 - **Числени типове**
 - **number**
 - **number(n)** (**float** при SQL/DS) за цели с променлива или фиксирана дължина.
 - **number(m,n)** (**Decimal** при SQL/DS) за реални числа с дължина *m* и *n* цифри след десетичната точка.
 - В ACCESS те са **Number(Integer, LongInteger, Byte, Single, Double)**, **Currency, Autonumber**

Богдан Шишеджиев - SQL

4

Дефиниция на домена

- Елементарни домени
 - **Date (Time** в SQL/DS) представя дата като сложна структура от полета.
 - В ACCESS те са **Date/Time**
 - **Raw** (ORACLE V6) за двоични данни, между които и bitmaps.
 - **BLOB** (Binary Large Objects).
 - В Access Мемо и **OLE** обекти
 - **Logical**.
- Именовани домени – в Interbase
 - CREATE DOMAIN

Богдан Шишеджиев - SQL

5

Дефиниция на схема

- Създаване на база от данни
 - **CREATE DATABASE** за създаване БД (не и в ACCESS)
 - **CREATE SCHEMA** [*SchemaName*] [[*authorisation*] *Authorization*] { *SchemaElementDefinilion* }
- Създаване на таблица
 - **CREATE TABLE** име (*Attribut1* *type1*, *Attribut2* *type2*, ...);
 - В ACCESS
 - CREATE TABLE** *table* (*field1* *type* [(*size*)] [NOT NULL] [*index1*] [, *field2* *type* [(*size*)] [NOT NULL] [*index2*] [, ...]] [, CONSTRAINT *multifieldindex* [, ...]])
 - В Interbase
 - CREATE TABLE** *table* [EXTERNAL [FILE] " <filespec>"] (<*col_def*> [, <*col_def*> | <*tconstraint*> ...]; <*col_def*> = *col* { *datatype* | COMPUTED [BY] (< *expr*>) | domain}
 - Дефиниция чрез заявка
 - CREATE TABLE** *nom* AS SELECT

Богдан Шишеджиев - SQL

6

Дефиниция на схема

- Дефиниция на стойност по подразбиране *default(expression)*
- Дефиниция на ограничения
 - UNIQUE
 - CONSTRAINT UNIQUE (имеатр,имеатр,...)
 - NOT NULL
 - Ключове
 - Първичен ключ
CONSTRAINT име ключ PRIMARY KEY (имеатр,имеатр,...)
 - Чужд ключ
CONSTRAINT име ключ FOREIGN KEY(имеатр, ...) REFERENCES име таблица (имеатр,...)
 - CHECK (израз)

Богдан Шишеджиев - SQL

7

Промяна на схема

- Премахване на схема
 - DROP TABLE *name*;
- Промяна на таблица
 - ALTER TABLE *name* ADD COLUMN(*name_of_column* *type*, ...);
 - добавяне на атрибут
 - ALTER TABLE *name* ADD CONSTRAINT ...;
 - ALTER TABLE *name* MODIFY(*column* *type*. ...); промяна на типа на атрибут
 - ALTER TABLE *name* DROP *column* ,, ...; премахване на атрибут

Богдан Шишеджиев - SQL

8

Индекси

- Създаване на индекс за бърз достъп по един или повече атрибути в таблица
CREATE [UNIQUE] INDEX name_index ON name_table (attribut [ASCIDESC],...);
- Премахване на индекса
DROP INDEX nom_index;

Богдан Шишеджиев - SQL

9

Изгледи

- Създаване на изглед
– CREATE VIEW name [(names of columns)] AS SELECT;
- Премахване на изглед
– DROP VIEW name ;
- Изгледът не съдържа физически данни. Той е само име на друга заявка. Неговото съдържание се изчислява при изпълнението на заявката. Имената на атрибутите не са необходими освен в случай на преименоване спрямо имената в съставлящите го таблици

Богдан Шишеджиев - SQL

10

ПРИМЕР

```
CREATE TABLE DEPT
(
  DEPTNO          INTEGER NOT NULL,
  DNAME           VARCHAR(14) CHARACTER SET ISO8859_1,
  LOC             VARCHAR(13) CHARACTER SET ISO8859_1,
  CONSTRAINT PK_DEPT PRIMARY KEY (DEPTNO)
);
CREATE TABLE EMP
(
  EMPNO           INTEGER NOT NULL,
  ENAME           VARCHAR(10) CHARACTER SET ISO8859_1,
  JOB             VARCHAR(9) CHARACTER SET ISO8859_1,
  MGR             INTEGER
  CHECK (MGR ISNULL or
        DeptNo = (select DeptNo from Employee E where E.RegNo = Superior) ,
  HIREDATE        TIMESTAMP,
  SAL             NUMERIC(9, 2),
  COMM            NUMERIC(9, 2),
  DEPTNO          INTEGER,
  CONSTRAINT PK_EMP PRIMARY KEY (EMPNO)
);
ALTER TABLE EMP ADD CONSTRAINT FK_DEPTNO FOREIGN KEY (DEPTNO)
REFERENCES DEPT (DEPTNO);
```

Богдан Шишеджиев - SQL

11

Примерна база от данни

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

EMP NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.1980	800		20
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30
7566	JONES	MANAGER	7839	02.4.1981	2975		20
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20
7839	KING	PRESIDENT		17.11.1981	5000		10
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30
7876	ADAMS	CLERK	7788	23.5.1987	1100		20
7900	JAMES	CLERK	7698	03.12.1981	950		30
7902	FORD	ANALYST	7566	03.12.1981	3000		20
7934	MILLER	CLERK	7788	23.1.1988	1300		10

Богдан Шишеджиев - SQL

12

Заявки

- Оператор SELECT

SELECT [DISTINCT | ALL] {* | израз| атрибут },...
FROM <table [alias]>,...

[WHERE условие за селекция или съединение]

[GROUP BY списък от атрибути]

[HAVING условие за селекция на група]

[UNION | INTERSECT | MINUS SELECT...]

[ORDER BY списък от атрибути [ASC | DESC]];

Богдан Шишеджиев - SQL

13

Проекция.

SELECT job, mgr FROM emp;
/* Query12 */

JOB	MGR
CLERK	7902
SALESMAN	7698
SALESMAN	7698
MANAGER	7839
SALESMAN	7698
MANAGER	7839
MANAGER	7839
ANALYST	7566
PRESIDENT	
SALESMAN	7698
CLERK	7788
CLERK	7698
ANALYST	7566
CLERK	7782

SELECT DISTINCT job, mgr
FROM emp; /* Query13 */

JOB	MGR
CLERK	7902
SALESMAN	7698
MANAGER	7839
ANALYST	7566
PRESIDENT	
CLERK	7788
CLERK	7698
CLERK	7782

Богдан Шишеджиев - SQL

14

Оператори в изразите

Код	Операция
+, -	Положително, обратен знак
*, /	Умножение, деление
+, -, (&)	Събиране, изваждане, конкатенация

SELECT ENAME || '(' || EMPNO || ')' 'NAME',
2 * SAL 'DoubleSal'
FROM EMP; (Query 39)

'NAME'	'DoubleSal'
SMITH(7369)	1600
ALLEN(7499)	3200
WARD(7521)	2500
JONES(7566)	5950
MARTIN(7654)	2500
BLAKE(7698)	5700
CLARK(7782)	4900
SCOTT(7788)	6000
KING(7839)	10000
TURNER(7844)	3000
ADAMS(7876)	2200
JAMES(7900)	1900
FORD(7902)	6000
MILLER(7934)	2600

Богдан Шишеджиев - SQL

15

Избор

SELECT * FROM emp WHERE deptno=10; /* Query14 */

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	09.6.1981	2450		10
7839	KING	PRESIDENT		17.11.1981	5000		10
7934	MILLER	CLERK	7782	23.1.1982	1300		10

Богдан Шишеджиев - SQL

16

Селекция

- Това е избор, следван от проекция
 SELECT ename, job, sal FROM emp WHERE job = 'MANAGER' AND sal>2500; /* Query15 */

ENAME	JOB	SAL
JONES	MANAGER	2975
BLAKE	MANAGER	2850

Богдан Шишеджиев - SQL

17

Операции

Код	Операция
+, -	Положително, обратен знак
*, /	Умножение, деление
+, -, (&)	Събиране, изваждане, конкатенация
=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN	сравнения
^, NOT	Степен, отрицание
AND	конюнкция
OR	дизюнкция

Богдан Шишеджиев - SQL

18

Предикати

BETWEEN / NOT BETWEEN
 SELECT ename FROM emp WHERE
 hiredate BETWEEN 1.1.81 AND
 31.12.81; /* Query1 */

IN / NOT IN
 SELECT ename FROM emp WHERE
 job In ('ANALYST','MANAGER');
 /* Query2 */

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
KING
TURNER
JAMES
FORD

ENAME
SMITH
JONES
BLAKE
CLARK
ADAMS
JAMES
MILLER

Богдан Шишеджиев - SQL

19

Предикати

- LIKE/NOT LIKE за сравнение на символни низове
 % замества 0 или повече символи, _ замества точно един символ.
 В ACCESS съответните символи са * и ?.

пример: LIKE 'TARKO%', LIKE '%WSKI', LIKE 'A_C'
 SELECT ename, job FROM Emp WHERE.ename LIKE "b*";/* Query4 */

ENAME	JOB
BLAKE	MANAGER

SELECT ename, job,sal FROM emp WHERE ename LIKE '%mi%';
 /*Query16*/

ename	job	sal
SMITH	CLERK	800
MILLER	CLERK	1300

Богдан Шишеджиев - SQL

20

UNION и други

Обединение на 2 заявки, връщаш резултат със същата: SELECT UNION SELECT
SELECT ename FROM query2 UNION ALL SELECT ename FROM query7 /* Query8 */

Сечение на 2 заявки с една и съща схема :

SELECT INTERSECTION SELECT ...

Разлика на 2 заявки с една и съща схема :

SELECT MINUS SELECT ...

```
ename  
SMITH  
JONES  
BLAKE  
CLARK  
ADAMS  
JAMES  
MILLER  
SMITH  
ADAMS  
JAMES  
MILLER
```

Функции

• Числови функции

- ABS(n) – Абсолютна стойност; CEIL(n) – най-малкото цяло, по-голямо от аргумента; FLOOR(n) – най-голямото цяло, по-малко от аргумента.
- ROUND(n[,m]) – закръгляване с *m* цифри след точката.
- TRUNC(n[,m]) – отрязване с *m* цифри след точката.
- MOD(n) – модул (остатък)
- POWER(m, n) – m^n , с *n* цяло.
- SQRT(n)

Функции

• Функции за обработка на символи.

- ASCII(char) връща ASCII кода на символа (ASC в ACCESS)
- CHR(n) връща символа, чийто ASCII код е параметър.
- INITCHAR(string) връща низа с като прави първата буква главна. (липсва в ACCESS)
- LENGTH(string) броя на символите в низ.
- LPAD(string, n, char), RPAD(string, n, char) запълва наляво или надясно низа с *n* екземпляра на символа *char*. (SPACES(n) връща *n* интервала в ACCESS)
- LTRIM(string, char), RTRIM(string, char) премахва всичко, което се намира наляво или надясно от символа *char*, ако той се намира в низа.
- TRANSLATE(string, c1, c2) замества в низа всички появявания на символа *c1* с *c2* (липсва в ACCESS).
- SUBSTR(string, pos, length) връща подниз със зададена дължина започващ от *pos*. MID\$(string, pos, length) в ACCESS)
- UPPER(string), LOWER(string) конвертира низа с главни или малки букви. (UCASE, LCASE в ACCESS)
- || е операция за конкатенация на низове. (В ACCESS (VB) операторът е &)

Функции

• Агрегатни функции.

- Те позволяват да се пресмятат обобщения за множества от кортежи
- COUNT(*) връща броя на кортежите; COUNT(attribute) : връща броя на стойностите на атрибута (без NULL стойностите).
- SUM(attribute) връща сумата на стойностите на атрибута.
- AVG(attribute) връща средната стойност на стойностите на атрибута.
- MIN(attribute), MAX(attribute) връщат минималната и максималната на стойност на атрибута
- VARIANCE(attribute) връща вариацията на стойностите на атрибута. (VAR en ACCESS)

Функции

- **Агрегатни функции.**

SELECT count(*) FROM emp WHERE deptno=20; /* Query10 */

Count
5

SELECT AVG(sal) average FROM emp WHERE deptno=20; /* Query11 */

Average
2175

SELECT count(Job) as Jobs FROM emp;
SELECT count(DISTINCT Job) as Jobs FROM emp;

Jobs
14

Jobs
5

Липсва в MS SQL

Богдан Шишеджиев - SQL

25

Функции

- **Функции на конвертиране на типовете**

- TO_CHAR(n [,format]) обръща число *n* в символен низ според зададения формат (STR\$ и FORMAT в ACCESS)
- TO_CHAR(date[, format]) обръща дата в низ (STR в ACCESS)
- TO_DATE(string,format) обръща низ в дата (CDATE в ACCESS)
- TO_NUMBER(string) обръща низ в цяло или реално число. (VAL в ACCESS)

Примери за формати : 'DD-MON-YY', 'DD/MM/YYYY', 'MONTH DD,YYYY'

- В Interbase функцията за конверсия е CAST(value AS datatype)

- **Други функции**

- DECODE(expression, v1,r1[,v2,r2[,v3,r3]]) връща r1, ако изразът expression има стойност, в противен случай връща r2, ако изразът е v2 и т.н.
- NVL(expression1, expression2) връща expression2, ако expression1 е NULL (NZ в ACCESS)
- GREATEST(e1, e2 ...), LEAST(e1, e2,...) връщат екстремалните стойности от списък.

- В Interbase се поддържат само функциите COUNT, SUM, CAST, AVG, UPPER, MAX, MIN

Богдан Шишеджиев - SQL

26

Съединение

- **Декартово произведение**

SELECT * FROM emp, dept; /* Query17 */

- **Съединение с квалификации**

SELECT * FROM emp,dept WHERE emp.deptno = dept.deptno; /* Query18 */
или на ACCESS или Interbase:

SELECT * FROM emp INNER JOIN dept ON emp.deptno = dept.deptno; /* Query19 */

SELECT * FROM emp E,dept D WHERE E.deptno = D.deptno; /*Query20 */
SELECT * FROM emp as E INNER JOIN dept as D ON E.deptno = D.deptno; /* Query21 */

Богдан Шишеджиев - SQL

27

Съединение

EMPNO	ENAME	JOB	MGR	HIRED	SAL	COMM	E.DEPTNO	D.DEPTNO	DNAME	LOC
7782	CLARK	MANAGER	7839	09.6.1981	2450		10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		17.11.1981	5000		10	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	23.1.1982	1300		10	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	17.12.1980	800		20	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	02.4.1981	2975		20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	19.4.1987	3000		20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	23.5.1987	1100		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	03.12.1981	3000		20	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	20.2.1981	1600	300	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	22.2.1981	1250	500	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	28.9.1981	1250	1400	30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	01.5.1981	2850		30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08.9.1981	1500		30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	03.12.1981	950		30	30	SALES	CHICAGO

Богдан Шишеджиев - SQL

28

Ограничение върху съединение

```
SELECT ename, job, dept.deptno, dname FROM emp, dept
WHERE emp.deptno = dept.deptno AND job = 'CLERK'; /* Query22 */
SELECT ename, job, dept.deptno, dname FROM emp INNER JOIN dept ON
emp.deptno = dept.deptno WHERE job = 'CLERK';
```

ename	job	deptno	dname
SMITH	CLERK	20	RESEARCH
ADAMS	CLERK	20	RESEARCH
JAMES	CLERK	30	SALES
MILLER	CLERK	10	ACCOUNTING

Богдан Шишеджиев - SQL

29

Подзаявки

```
SELECT ename, deptno FROM emp
WHERE deptno IN ( SELECT deptno FROM dept WHERE dname LIKE '%S%' )
; /* Query23 */
SELECT ename, e.deptno FROM emp E INNER JOIN dept D ON
e.deptno=d.deptno WHERE dname LIKE '%S%';
```

ename	deptno
SMITH	20
ALLEN	30
WARD	30
JONES	20
MARTIN	30
BLAKE	30
SCOTT	20
TURNER	30
ADAMS	20
JAMES	30
FORD	20

Богдан Шишеджиев - SQL

30

Подзаявки

- **SOME, ANY**

```
SELECT ename FROM emp WHERE
sal > ANY (SELECT sal FROM emp
WHERE deptno = 20); /* Query5 */
SELECT ename FROM emp WHERE
sal > (SELECT MIN(sal) FROM emp
WHERE deptno = 20);
```

ENAME
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
JAMES
FORD
MILLER

- **ALL**

```
SELECT ename FROM emp WHERE
sal > ALL (SELECT sal FROM emp
WHERE deptno = 20); /* Query6 */
SELECT ename, sal FROM emp
WHERE deptno = 10 AND sal > (
SELECT MAX(sal) FROM emp
WHERE deptno = 20); /* Query24 */
```

ename	sal
KING	5000

- **EXISTS**

```
SELECT dname, deptno FROM dept
WHERE EXISTS (SELECT * FROM
emp WHERE dept.deptno =
emp.deptno); /* Query 31 */
```

dname	deptno
ACCOUNTING	10
RESEARCH	20
SALES	30

Богдан Шишеджиев - SQL

31

Подзаявки

- **Корелативни подзаявки**

```
SELECT ename, deptno FROM emp E1 WHERE E1.sal > (SELECT AVG(E2.sal)
FROM emp E2 WHERE E1.deptno=E2.deptno); /* Query25 */
```

- **Подзаявки 2-ро ниво**

```
SELECT dname, deptno FROM dept AS D2 WHERE EXISTS (SELECT * FROM
emp E, dept D WHERE E.deptno=D.Deptno and D2.deptno = D.deptno and
E.sal > (SELECT AVG(sal) FROM emp E1, dept D1 WHERE
E1.deptno=D1.deptno AND D1.dname LIKE 'ACCOUNTING')) /* Query32 */;
```

ename	deptno
ALLEN	30
JONES	20
BLAKE	30
SCOTT	20
KING	10
FORD	20

dname	deptno
ACCOUNTING	10
RESEARCH	20

Богдан Шишеджиев - SQL

32

Групиране

- GROUP BY позволява да се групират резултатите според даден критерий и кортежите от всяка група да се обработват от агрегатни функции. Тази клауза се прилага за тези атрибути, които не са параметри на агрегатна функция!!
- Редът на изпълнение е следния:
 - Ако има клауза WHERE СУБД премахва всички кортежи, които не удовлетворяват условието.
 - Извършва групирането и изчислява агрегатните стойности.
 - Премахва всички групи, които не удовлетворяват условието в клаузата HAVING.

```
SELECT deptno, MIN(sal), MAX (sal) FROM emp GROUP BY deptno; /* Query26 */
```

```
SELECT deptno, MIN(sal), MAX (sal) FROM emp WHERE job = 'CLERK' GROUP BY deptno; /* Query27 */
```

deptno	Minsal	Maxsal
10	1300	5000
20	800	3000
30	950	2850

Богдан Шишеджиев - SQL

deptno	Minsal	Maxsal
10	1300	1300
20	800	1100
30	950	950

Богдан Шишеджиев - SQL

35

Групиране

• HAVING

позволява да се изразят критерии спрямо групите (агрегатните операции) и се използва само с GROUP BY

```
SELECT deptno, MIN(sal), MAX(sal) FROM emp GROUP BY deptno HAVING MAX(sal) >1200; /* Query28 */
```

```
SELECT deptno, MIN(sal), MAX(sal) FROM emp WHERE job = 'CLERK' GROUP BY deptno HAVING MIN(sal)<1000; /* Query29 */
```

deptno	Minsal	Maxsal
10	1300	5000
20	800	3000
30	950	2850

deptno	Minsal	Maxsal
20	800	1100
30	950	950

Богдан Шишеджиев - SQL

34

Сложен пример

- Да се намери частта от броя на служителите и на заплатите за всеки отдел

– ORACLE

```
SELECT a.deptno "Department", a.num_emp/b.total_count "%Employees", a.sal_sum/b.total_sal "%Salary"
```

```
FROM (SELECT deptno, COUNT(*) num_emp, SUM(SAL) sal_sum FROM emp
```

```
GROUP BY deptno) a,
```

```
(SELECT COUNT(*) total_count, SUM(sal) total_sal FROM emp) b ;
```

Богдан Шишеджиев - SQL

35

Сложен пример

```
CREATE VIEW X AS SELECT deptno, COUNT(*) num_emp, SUM(SAL) sal_sum FROM emp GROUP BY deptno; /*Query35*/
```

```
CREATE VIEW Y AS SELECT COUNT(*) total_count, SUM(sal) total_sal FROM emp); /*Query36*/
```

```
SELECT x.deptno AS Department, x.num_emp/y.total_count AS Pr_Employees, x.sal_sum/y.total_sal AS Pr_Salary FROM X, Y; /* Query37 */
```

deptno	num_emp	sal_sum
10	3	8750
20	5	10875
30	6	9400

total_count	total_sal
14	29025

Department	Pr_Employees	Pr_Salary
10	0.214285714285714	0.301464254952627
20	0.357142857142857	0.374677002583979
30	0.428571428571429	0.323858742463394

Богдан Шишеджиев - SQL

36

Външно съединение

- Синтаксис

FROM table ...{LEFT | RIGHT | FULL } [OUTER]; JOIN table ON лог.израз

- Пример

```
SELECT emp.ENAME, emp.JOB, dept.DEPTNO, dept.DNAME
FROM emp RIGHT JOIN dept ON
emp.DEPTNO = dept.DEPTNO;

SELECT dept.DNAME,
       Count(emp.EMPNO) AS
       CountOfEMPNO
FROM emp RIGHT JOIN dept ON
emp.DEPTNO = dept.DEPTNO
GROUP BY dept.DNAME; /*Query 34*/
```

ENAME	JOB	DEPTNO	DNAME
JONES	MANAGER	20	RESEARCH
SCOTT	ANALYST	20	RESEARCH
ADAMS	CLERK	20	RESEARCH
FORD	ANALYST	20	RESEARCH
ALLEN	SALESMAN	30	SALES
WARD	SALESMAN	30	SALES
MARTIN	SALESMAN	30	SALES
BLAKE	MANAGER	30	SALES
TURNER	SALESMAN	30	SALES
JAMES	CLERK	30	SALES
		40	OPERATIONS

DNAME	CountOfEMPNO
ACCOUNTING	3
OPERATIONS	0
RESEARCH	5
SALES	6

37

Подреждане на кортежите

- ORDER BY

ORDER BY {ASC | DESC}

- Пример

```
SELECT ename, deptno, sal FROM emp ORDER BY deptno, sal DESC; /*
Query30 */
```

ename	deptno	sal
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000
SCOTT	20	3000
JONES	20	2975
ADAMS	20	1100
SMITH	20	800
BLAKE	30	2850
ALLEN	30	1600
TURNER	30	1500
MARTIN	30	1250
WARD	30	1250

Богдан Шишеджиев - SQL

38

Йерархия

- Изобразяване на йерархични данни

```
SELECT [level] ... CONNECT BY PRIOR expr = expr START WITH expr
SELECT Lpad(' ', 2*level)|| nom, num FROM EMP CONNECT BY PRIOR
chef= num START WITH nom='A';
```

```
SELECT LPAD(' ', 2*(LEVEL-1)) || ename, empno, mgr, job
FROM emp START WITH job = 'PRESIDENT'
CONNECT BY PRIOR empno = mgr;
```

NUM	NOM	CHEF
1	A	NULL
2	D	NULL
3	H	2
4	B	1
5	E	2
6	F	5
7	C	1
8	G	5

A 1

B 4

C 7

D 2

E 5

F 6

G 8

H 3

Богдан Шишеджиев - SQL

39

Йерархия

- В примерната база

```
SELECT LPAD(' ', 2*(LEVEL-1)) || ename, empno, mgr, job
FROM emp START WITH job = 'PRESIDENT'
CONNECT BY PRIOR empno = mgr;
```

ENAME	EMP NO	MGR	JOB
KING	7839		PRESIDENT
BLAKE	7698	7839	MANAGER
ALLEN	7499	7698	SALESMAN
WARD	7521	7698	SALESMAN
MARTIN	7654	7698	SALESMAN
TURNER	7844	7698	SALESMAN
JAMES	7900	7698	CLERK
CLARK	7782	7839	MANAGER
MILLER	7934	7782	CLERK
JONES	7566	7839	MANAGER
SCOTT	7788	7566	ANALYST
ADAMS	7876	7788	CLERK
FORD	7902	7566	ANALYST
SMITH	7369	7902	CLERK

Богдан Шишеджиев - SQL

40

Заявки с параметри

- Параметри към заявките - :varname
- Access – property *parameters*

Богдан Шишеджиев - SQL

41

Обновяване

- **Вмъкване**

INSERT INTO table [(col1[,col2...])] VALUES(списък от стойности); или :

INSERT INTO table [(col1[,col2...])] VALUES SELECT ... ;

– Пример :

INSERT INTO Emp (EMPNO, ENAME, JOB, HIREDATE, SAL, COMM, DEPTNO)

SELECT [EMPNO]+20 AS Expr5, ENAME, "CLERK" AS Expr2, #9/1/99# AS Expr4, 800 AS Expr3, COMM, 40 AS Expr1 FROM Emp WHERE DEPTNO=10;

Богдан Шишеджиев - SQL

42

Обновяване

- **Изменение**

UPDATE table или view SET {column = expression | (list of columns)=(list of expressions)} [WHERE condition] ;

– Изразът може да бъде SELECT оператор, който изработва необходимите стойности.

– Пример:

UPDATE Emp SET Emp.SAL = [Sal]+100 WHERE DEPTNO=40;

- **Изтриване**

DELETE FROM table [WHERE condition];

– Пример:

DELETE FROM emp WHERE DEPTNO=40;

Богдан Шишеджиев - SQL

43

Управление на правата на достъп

- **Общи права на достъп:**

– GRANT CONNECT | RESOURCE | ROLE TO потребител IDENTIFIED BY парола;

- CONNECT, RESOURCE са предефинирани роли в СУБД Oracle. Тяхното използване не се препоръчва в последната версия..

– ROLE е именувано множество от привилегии които могат да се дават като едно цяло. То се обработва от командите:

- CREATE ROLE
- ALTER ROLE
- SET ROLE

Богдан Шишеджиев - SQL

44

Управление на правата на достъп

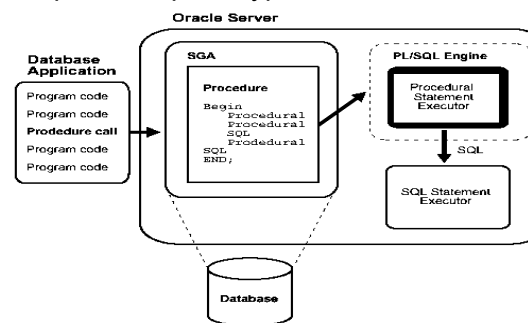
- Достъп до обекти (с евентуално право за препредаване) :
 - GRANT SELECT | INSERT | DELETE | UPDATE | ALTER | INDEX | CLUSTER ON таблица или изглед TO потребител или роля [WITH GRANT OPTION] ;
 - GRANT OPTION дава на потребителя право да предава собствените си права на други потребители.
- Отнемане на права
 - REVOKE отнема права на потребителите
 - REVOKE право ON таблица или изглед FROM потребител

Богдан Шишеджиев - SQL

45

Други средства

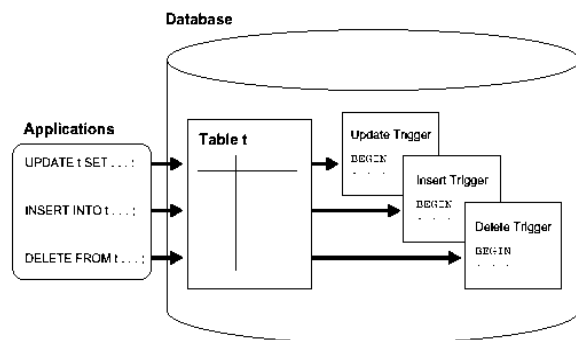
- Курсори
 - Указатели към множества от кортежи, използвани като записи
- Съхранени процедури



46

Други средства

- Тригери



Богдан Шишеджиев - SQL

47

Вградени езици

- PL/SQL – Oracle
- VBA – ACCESS
- VB, C# - SQL Server

Богдан Шишеджиев - SQL

48

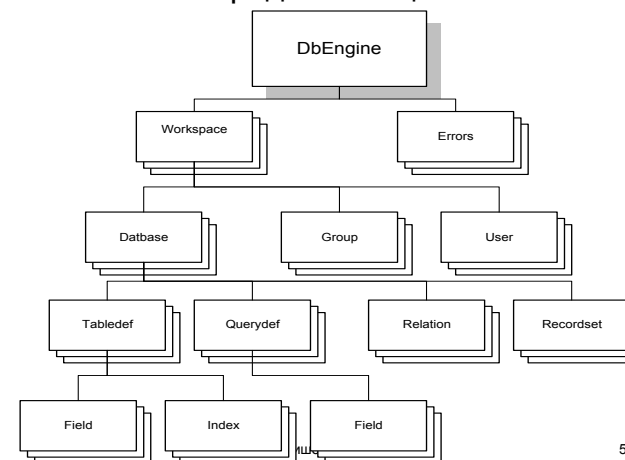
Транзакции

- Дефиниция – последователност от действия, които се възприемат и изпълняват като едно цяло
- Оператори за управление на транзакциите
 - begin trans
 - commit
 - rollback

Богдан Шишеджиев - SQL

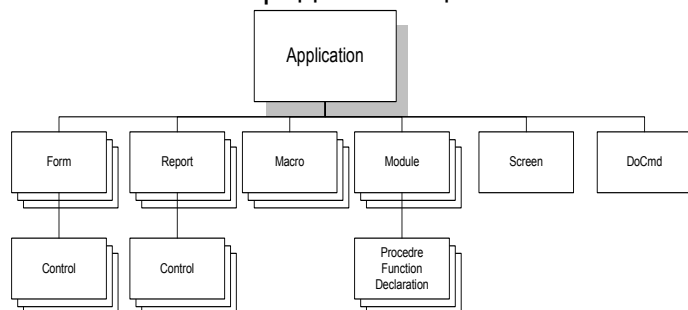
49

Вградени езици



50

Вградени езици



Богдан Шишеджиев - SQL

51

Упражнение

• Схема

Нека е дадена реляционната схема:

Supplier(SupNo, Name, Country, Address, TelNo)

Article(ArtNo, Name, Unit)

Price(SupNo, ArtNo, PriceUn)

Delivery(LivNo, Date, SupNo)

Detail(LivNo, ArtNo, Quant)

Да се напишат операторите SQL. Даващи отговор на следните въпроси:

1. Кои са доставчиците с имена съдържащи "ma"?

```
SELECT * FROM Supplier WHERE Name LIKE '%ma%'
```

2. Какви са минималните, максималните и средните цени за всяка стока?

```
SELECT A.NAME, Max(PriceUn), Min(PriceUn), Avg(PriceUn)
FROM Price P INNER JOIN Article A ON P.ArtNo = A.ArtNo;
GROUP BY A.NAME;
```

3. Кои са стоките доставяни от американски доставчици?

```
SELECT DISTINCT A.NAME FROM Supplier S, Price P, Article A
WHERE A.ArtNo=P.ArtNo and S.SupNo=P.SupNo and S.Country='USA';
SELECT DISTINCT NAME FROM Price P, Article A
WHERE A.ArtNo=P.ArtNo and P.SupNo in
(SELECT S.SupNo FROM Supplier S WHERE Country='USA');
```

52

Упражнение

Supplier(SupNo, Name, Country, Address, TelNo)
Article(ArtNo, Name, Unit)
Price(SupNo, ArtNo, PriceUn)
Delivery(LivNo, Date, SupNo)
Detail(LivNo, ArtNo, Quant)

4. Какъв е броя на стоките доставян от всеки доставчик?

```
SELECT S.SupNo, S.Name, Count(ArtNo)
FROM Supplier S LEFT JOIN Price P ON S.SupNo = P.SupNo
GROUP BY S.SupNo, S.Name;
```

5. Кой доставчик не доставя никакъв вид сирене (всички имена на сирена съдържат името "сирене")

```
SELECT S.SupNo S.Name FROM Supplier S, Article A, Price P
WHERE S.SupNo=P.SupNo and A.ArtNo=P.ArtNo and
NOT (A.Name LIKE '%сирене%');
```

```
SELECT S.SupNo, Name FROM Supplier S
WHERE S.SupNo Not IN (SELECT P.SupNo FROM Article A, Price P
WHERE A.ArtNo=P.ArtNo and A.Name LIKE '%сирене%');
```

Богдан Шишеджиев - SQL

53

Упражнение

Supplier(SupNo, Name, Country, Address, TelNo)
Article(ArtNo, Name, Unit)
Price(SupNo, ArtNo, PriceUn)
Delivery(LivNo, Date, SupNo)
Detail(LivNo, ArtNo, Quant)

6. Направете сравнение между средните стойности на всяка стока доставяни от американските и останалите доставчици

```
CREATE VIEW USA (Name, avg) AS
SELECT A.Name, Avg(PriceUn)
FROM (Supplier S INNER JOIN Price P ON S.SupNo=P.SupNo)
JOIN Article A ON P.ArtNo = A.ArtNo
WHERE S.Country LIKE 'USA'
GROUP BY A.Name;
```

```
CREATE VIEW NoUSA (Name, avg) AS
SELECT A.Name, Avg(PriceUn)
FROM (Supplier S INNER JOIN Price P ON S.SupNo=P.SupNo)
JOIN Article A ON P.ArtNo = A.ArtNo
WHERE NOT S.Country LIKE 'USA'
GROUP BY A.Name;
```

```
SELECT U.Name, U.Avg as USA, N.Avg AS NoUSA FROM USA U, NOUSA N
WHERE U.Name = N.Name;
```

```
SELECT NVL(U.Name,N.Name), U.Avg as USA, N.Avg AS NoUSA
FROM USA U FULL JOIN NOUSA N ON U.Name = N.Name;
```

Упражнение

Supplier(SupNo, Name, Country, Address, TelNo)
Article(ArtNo, Name, Unit)
Price(SupNo, ArtNo, PriceUn)
Delivery(LivNo, Date, SupNo)
Detail(LivNo, ArtNo, Quant)

7. Направете месечен отчет по доставчици – всеки доставчик стоки на каква сума е доставил?

```
SELECT S.SupNo, S.Name, Sum(Quant*PriceUn)
FROM Supplier S, Price P, Delivery D, Detail L
WHERE S.SupNo = P.SupNo and S.SupNo = D.SupNo and D.LivNo= L.LivNo and
P.ArtNo=L.ArtNo and Date Between 1.4.03 and 30.4.03
GROUP BY S.SupNo, S.Name
ORDER By S.Name;
```

8. Намерете средната сума на дневните доставки за последния месец.

```
CREATE VIEW Day AS SELECT Date, Sum(Quant*PriceUn) AS Suma
FROM Price P, Delivery D, Detail L
WHERE D.LivNo= L.LivNo and P.ArtNo=L.ArtNo
and Date Between 1.4.03 and 30.4.03
GROUP BY Date;
SELECT AVG(Suma) FROM Day;
```

Богдан Шишеджиев - SQL

55

Пример 2

За релационната схема

Jet (JetNum, JetName, Cap) JetName е името на модела на самолета напр. Airbus 320A, а Cap е капацитетът му.

Pilotes (NumPil, Name, Birth,City).

Fly(FlyNum, CityL, CityA, DateL, DateA, NumPil, JetNom, Price) Price е минимална цена на място за този полет.

Class (JetNum,Class, CoeffPlace, CoeffPrice) CoeffPlace, в интервала [0, 1], дава процента на местата в дадената класа за съответния самолет. CoeffPrice (>=1) дава множител, с който се умножава Price, за да се получи реалната цена за полета в исканата класа.

Clients(NumCl, NameCl, Street, StrNum, PostCode, CityCl)

Reservations(NumCl, FlyNum, Class, NbPlaces)

Богдан Шишеджиев - SQL

56

Пример 2

Да се напишат SQL оператори, които отговарят на следните въпроси

1. Имената на пилотите, които управляват всички Boeing-и.
2. Номерата на клиента, които са направили повече от 3 резервации и общата сума на резервациите за всеки от тях.
3. Номерата и имената на клиентите, които са резервирали място в полет превозващ г-н Алабашев.
4. Номера и името на най-възрастният пилот управляващ Airbus.
5. Броят на градовете (на пристигащите), обслужвани от компанията.
6. Номерата на полета, които осигуряват обратната траектория на полет F101.
7. Номерата на пилотите, които не осъществяват никакви полети. (2 начина).
8. Увеличете цените на всички полети от София с 10%.
9. Да се намерят най-печелившите направления
10. Да се намерят най-лоялните клиенти. (Платили най-много)