

SQL

A Query Language for Relational Databases

John Sieg, UMass Lowell

History

- Structured Query Language, created in 1970s at IBM Research Labs, San Jose
- Has evolved, acquiring more and more features
- SQL 92 is widely supported at various levels.
- SQL:1999 (ANSI and ISO) is current standard.

John Sieg, UMass Lowell

A Sample Database

student	sid	sname	yog	gpa	sex	major
	s1	Larry	grad	3.5	M	cs
	s2	Moe	2005	3.2	M	math
	s3	Curly	2004	4	M	cs
	s4	Shemp	2000		M	physics

course	cid	cname	description
	c1	Database I	the best course ever
	c2	Database II	the second best course ever
	c3	Compilers	
	c4	Poetry	

John Sieg, UMass Lowell

A Sample Database, cont.

course_off	cid	sec_no	semest	instruct	room	time_slot
	c1	201	f2005	f1	Olsen 420	T 5-7:30
	c1	201	f2004	f1	Olsen 999	T 5-7:30
	c2	202	s2005	f1	Ball 210	M 5-7:30
	c2	202	s2000	f2	Ball 222	M 5-7:30
	c3	201	f2005	f3	Olsen 115	M 3-4:30
	c4	201	s2005	f4	Ball 333	Th 8-10

enr	sid	cid	sec_no	semest	grade
	s1	c1	201	f2004	a
	s1	c2	202	s2005	b
	s2	c1	201	f2004	a
	s2	c2	202	f2005	b
	s2	c3	201	f2005	a
	s2	c4	201	s2005	c
	s3	c1	201	f2005	a
	s3	c2	202	s2000	a

CREATE TABLE

```
CREATE TABLE student
(
  sid      CHAR(5),
  sname    CHAR(16),
  yog      CHAR(4),
  gpa      DECIMAL(3,2),
  sex      CHAR(1),
  major    CHAR(16),
  PRIMARY KEY (sid)
)
```

John Sieg, UMass Lowell

CREATE TABLE, cont'd

```
CREATE TABLE enr
(
  sid      CHAR(5),
  cid      CHAR(5),
  sec_no   CHAR(5),
  semester CHAR(4),
  grade    CHAR(2),
  PRIMARY KEY (sid, cid, sec_no, semester),
  FOREIGN KEY (sid) REFERENCES student,
  FOREIGN KEY (cid, sec_no, semester)
  REFERENCES course_off
)
```

John Sieg, UMass Lowell

Domains

```
CREATE DOMAIN SID_DOMAIN CHAR(5);
CREATE DOMAIN SNAME_DOMAIN CHAR(16);
...
CREATE TABLE student
    (sid      SID_DOMAIN,
     sname    SNAME_DOMAIN,
     ...)
```

John Sieg, UMass Lowell

Domain Constraints

```
CREATE DOMAIN SEX_DOMAIN CHAR(1)
CONSTRAINT CHECK_SEX
CHECK
    ( VALUE IN ('m', 'f'))
```

John Sieg, UMass Lowell

DROP TABLE

```
DROP TABLE student
```

removes the *student* table,
including schema

John Sieg, UMass Lowell

Indices (not part of SQL standards)

```
CREATE CLUSTERED INDEX StudentInd
ON student (sid)
    builds an index on student.sid
```

```
DROP student.StudentInd
    deletes the index
```

John Sieg, UMass Lowell

Updates

```
INSERT INTO student (sid, sname, yog, gpa,
sex, major)
VALUES
    ('s1', 'Larry', 'grad', 4.0, 'm', 'cs');
```

```
UPDATE student
SET major = 'cs'
WHERE student.gpa = 4.0;
```

```
DELETE FROM student
WHERE yog = '2005';
```

John Sieg, UMass Lowell

Syntax - The SELECT Statement

```
SELECT column_list
FROM tablename_list
[ WHERE search_condition ]
[ GROUP BY column_list ]
[ HAVING group_condition ]
[ ORDER BY column_list ]
```

John Sieg, UMass Lowell

SQL Select Statement

Give the names of Computer Science (CS) majors.

```
SELECT sname
FROM student
WHERE major = 'cs'
```

sname
Larry
Curly

John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the names of students who have received at least one grade of 'A.'

```
SELECT student.sname
FROM student, enr
WHERE enr.grade = 'a'
AND student.sid = enr.sid
```

sname
Larry
Moe
Curly

John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the names of students who have received at least one grade of 'A'.

```
SELECT DISTINCT student_sname
FROM student, enr
WHERE enr.grade = 'a'
AND student.sid = enr.sid
```

sname
Curly
Larry
Moe

John Sieg, UMass Lowell

SQL UNIONS

Give the CIDs of courses that either have prerequisites or are prerequisites for other courses or both.

```
SELECT cid FROM prereq
UNION
SELECT prereq_cid FROM prereq
```

- Duplicates not returned. UNION ALL returns duplicates.
- SQL also has INTERSECT and EXCEPT operators.
- Union compatible => same number and types of columns

John Sieg, UMass Lowell

Table Aliases

Give the names of courses that have prerequisites and the names of the prerequisites.

```
SELECT c1.cname AS courseName,
       c2.cname AS prereq
FROM course c1, prereq, course c2
WHERE c1.cid = prereq.cid
AND prereq.prereq_cid = c2.cid
```

John Sieg, UMass Lowell

SQL Views

Create a view that joins together students with their enrollments.

```
CREATE VIEW student_enr
(sid, sname, major, yog, cid, sec_no,
 semester, grade)
AS
SELECT sid, sname, major, yog, enr.cid,
       sec_no, semester, grade
FROM student, enr
WHERE student.sid = enr.sid
```

John Sieg, UMass Lowell

Querying a View

Give the names of students who have received at least one grade of 'A'.

```
SELECT sname
FROM student_enr
WHERE grade = 'A'
```

John Sieg, UMass Lowell

Pattern Matching

Give IDs of courses named "Database something-or-other".

```
SELECT cid
FROM course
WHERE cname LIKE 'Database%'
```

John Sieg, UMass Lowell

Pattern Matching, cont.

Pattern	Meaning	Example
%	0 or more characters	'John%' (a string beginning with 'John')
_	any one character	'a_' ('a0' or 'a1' or ... 'aa' or ...)
[]	one character in the brackets	'a[bcd]' ('ab' or 'ac' or 'ad')

John Sieg, UMass Lowell

Nested SQL Select Statements

Give the names of students who have enrolled in course 'c1'.

```
SELECT sname
FROM student
WHERE sid IN
  (SELECT sid
   FROM enr
   WHERE cid = 'c1')
```

John Sieg, UMass Lowell

Nested SQL Select Statement, cont.

Give the names of students who have the highest GPA.

```
SELECT sname
FROM student
WHERE gpa =
  (SELECT MAX(gpa)
   FROM student)
```

John Sieg, UMass Lowell

Nested SQL Select Statement, cont.

Give the names of students who have received at least one grade of 'A'.

```
SELECT sname
FROM student
WHERE 'A' IN
  (SELECT grade
   FROM enr
   WHERE student.sid = enr.sid)
```

John Sieg, UMass Lowell
John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the names of students who have received at least one grade of 'A'.

```
SELECT sname
FROM student
WHERE EXISTS
  (SELECT *
   FROM enr
   WHERE student.sid = enr.sid
   AND enr.grade = 'a')
```

John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the SIDs of students who have received A's in all their courses.

```
SELECT sid
FROM student
WHERE NOT EXISTS
  (SELECT *
   FROM enr
   WHERE enr.sid = student.sid
   AND enr.grade <> 'a')
```

John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the names of students who have enrolled in every course.

```
SELECT sname
FROM student
WHERE NOT EXISTS
  (SELECT *
   FROM course
   WHERE NOT EXISTS
     (SELECT *
      FROM enr
      WHERE student.sid = enr.sid
      AND enr.cid = course.cid))
```

John Sieg, UMass Lowell

GROUP BY

Give each (enrolled) student's sid and number of courses he has been enrolled in.

```
SELECT sid,
       COUNT(DISTINCT cid) AS num
FROM enr
GROUP BY sid
```

John Sieg, UMass Lowell

HAVING

Give the SIDs of students who have been enrolled in more than two different courses.

```
SELECT sid
FROM enr
GROUP BY sid
HAVING COUNT(DISTINCT cid) > 2
```

John Sieg, UMass Lowell

SQL Select Statement, cont.

Give the names of students who have received at least one grade of 'A'.

```
SELECT sname
FROM student JOIN enr
ON student.sid = enr.sid
WHERE enr.grade = 'a'
```

John Sieg, UMass Lowell

Outer Joins

Tell me everything about students and their enrollments. Include students who don't have any enrollments.

```
SELECT *
FROM student LEFT OUTER JOIN enr
ON student.sid = enr.sid
```

John Sieg, UMass Lowell

A Left Outer Join

st.sid	sname	yog	gpa	sex	major	enr.sid	cid	sec	sem	grade
s1	Larry	grad	3.5	M	cs	s1	c1	201	f2004	a
s1	Larry	grad	3.5	M	cs	s1	c2	202	f2005	b
s2	Moe	2005	3.2	M	math	s2	c1	201	f2004	a
s2	Moe	2005	3.2	M	math	s2	c2	202	f2005	b
s2	Moe	2005	3.2	M	math	s2	c3	201	f2005	a
s2	Moe	2005	3.2	M	math	s2	c4	201	s2005	c
s3	Curly	2004	4	M	cs	s3	c1	201	f2005	a
s3	Curly	2004	4	M	cs	s3	c2	202	s2000	a
s4	Shemp	2000		M	physics					

John Sieg, UMass Lowell

Cartesian Products

Show me all pairings of students with enrollments, whether related or not.

```
SELECT sname
FROM student CROSS JOIN enr
```

John Sieg, UMass Lowell

Triggers / Active Databases

```
CREATE TRIGGER RejectPoorStudents
AFTER INSERT ON Student /* Event */
WHEN Student.salary=0 /* Cond'n */
FOR EACH ROW
INSERT INTO RejectLog /* Action */
VALUES (sid, sname, sex)
```

John Sieg, UMass Lowell

Embedded SQL: Declaring a Cursor

```
DECLARE student_cursor CURSOR
FOR
SELECT S.sname, S.yog
FROM student S
WHERE S.gpa > :ext_gpa;
```

John Sieg, UMass Lowell

Embedded SQL: Retrieving from a Cursor

```
OPEN student_cursor
...
FETCH student_cursor
INTO :ext_sname, :ext_yog
...
CLOSE student_cursor
```

John Sieg, UMass Lowell

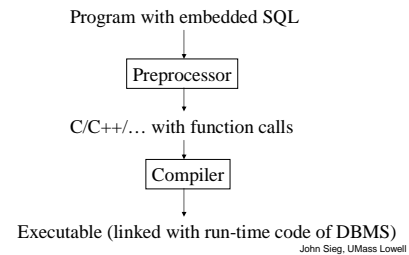
Dynamic SQL

```
char user_query[];
// Build an SQL query
// dynamically.
getquery(user_query);

EXEC SQL PREPARE SqlStatement
FROM :user_query;
EXEC SQL EXECUTE SqlStatement
```

John Sieg, UMass Lowell

Translating a Program with Embedded SQL



John Sieg, UMass Lowell

SQLJ Example

```
String sid;
// Missing code: Get sid from user.
#sql iterator StudIteratorClass
(sname, gpa);
StudIteratorClass StudentIter;
#sql StudentIter =
{SELECT sname, gpa
FROM student
WHERE sid = :sid};
```

John Sieg, UMass Lowell

SQLJ Example, cont'd

```
while (StudentIter.next())
{
    System.out.println(
        StudentIter.sname() + " "
        + StudentIter.gpa());
}
StudentIter.close();
```

John Sieg, UMass Lowell

Competing Standard APIs: ODBC and JDBC

- Function calls are standardized, unlike embedded SQL.
- Access to different data sources is mediated by ODBC/JDBC drivers.
- Unlike with embedded SQL or SQLJ, binding to database system is done at run-time.

John Sieg, UMass Lowell

Conclusions

- SQL 92 is an ugly language. Alternatives such as Quel are cleaner.
- SQL:1999 cleans up some of the orthogonality issues and adds many new features. We'll see some of the new features of SQL:1999 later.
- SQL will continue to grow. Current work includes extensions to query XML data.

John Sieg, UMass Lowell