SQL:1999 and SQL:2003

- Previously known as SQL3
- Support for recursion, OLAP, and everything else
- Most important extensions deal with user-defined types, complex types, inheritance, and other "object-oriented" features

User-Defined Types in SQL:1999

```sql
CREATE TYPE StudentType AS
({
  sid CHAR(9),
sname CHAR(32),
yog CHAR(4),
... CREATE METHOD is_grad RETURNS BOOLEAN
  BEGIN IF yog='grad' THEN RETURN TRUE;
    ELSE RETURN FALSE;
  END
})
(external functions also supported)
```

User-Defined Types in SQL:1999, cont’d

```sql
CREATE TABLE student
OF StudentType
(
  PRIMARY KEY (sid)
)
```
SELECT * FROM student S WHERE is_grad(S)

CREATE TYPE TranscriptRowType AS (cid CHAR(6), sec_no CHAR(3), semester CHAR(20), grade NUMBER);
CREATE TYPE transcriptType AS TABLE OF TranscriptRowType;

CREATE TYPE StudentType AS (sid CHAR(9), sname VARCHAR2(20), major VARCHAR2(20) ARRAY[5], yog CHAR(4), gpa NUMBER, transcript transcriptType, ...

CREATE TYPE StudentType AS (sid CHAR(9), sname VARCHAR2(20), major VARCHAR2(20) ARRAY[5], yog CHAR(4), gpa NUMBER, transcript transcriptType, ...

CREATE TABLE student OF StudentType ...

SELECT S.sname FROM student AS S WHERE S.major[1]="Computer Science";
SELECT S.sname, M.mname FROM student AS S, UNNEST S.major AS M (mname)
Table Inheritance in SQL:1999

```
CREATE TABLE ta
UNDER student
(
    salary DECIMAL(5, 2)
)
```

Type inheritance for “non-final” types also supported, with possible overriding of methods.

Recursion in SQL:1999

```
WITH RECURSIVE
prereq_tc (cid, prereq_cid) AS
(SELECT cid, prereq_cid
FROM prereq
UNION
SELECT T.cid, P.prereq_cid FROM prereq_tc T, prereq P
WHERE T.prereq_cid = P.cid)
```

Recursion in SQL:1999, cont'd

<table>
<thead>
<tr>
<th>prereq</th>
<th>cid</th>
<th>prereq_cid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c2</td>
<td>c1</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>c2</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>c3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>prereq_tc</th>
<th>cid</th>
<th>prereq_cid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c2</td>
<td>c1</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>c2</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>c3</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>c1</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>c2</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>c1</td>
</tr>
</tbody>
</table>

SQL:1999 – Other New Features

- ref types (= object identity)
- Strong typing
- OLAP (CUBE, ROLLUP, expressions in ORDER BY)
- Triggers
- SQL routines and external routines
- Stored Procedures
- Savepoints
- SQLJ
SQL:1999 – Other New Features, cont’d

• Control constructs (IF, CASE, LOOP, WHILE, FOR, LOOP, …)
• New Types: Boolean, LOBs (binary & character) with locators to navigate through huge objects
• Roles (for security)
• Object Views, Typed views, View hierarchies
• Transform functions (from_sql, to_sql methods)
• Standard type libraries (e.g., SQL/MM Full text, SQL/MM Spatial, SQL/MM Still Image)
• SQL/MED (Management of External Data)
• more orthogonality

SQL:2003

• Previously known as SQL 4
• Major new functionality is XML support
• Cleans up SQL:1999

SQL:2003: Mapping Relations to XML Documents

```xml
<STUDENT>
  <row>
    <SID> s1 </SID> <SNAME>Larry</SNAME> ...
  </row>
  <row> ... </row> ...
</STUDENT>
```

also mapping of SQL types to XML schema definitions

SQL:2003: XML Data Type

```sql
SELECT XMLELEMENT
  (NAME "student",
   XMLELEMENT
     (NAME "sid", s.sid)
   XMLELEMENT
     (NAME "sname", s.sname))
FROM student s
returns

<student>
  <sid> s1 </sid> <sname>Larry</sname>
</student>
<student>
  <sid> s2 </sid> <sname>Moe</sname>
</student>
...
```
SQL:2003: Other New Features

- Multiset type constructor
- Sequence generators
- Generated (computed) attributes
- Table functions (functions that compute tables)
- MERGE (merge “transaction” tuples into another table via INSERT or UPDATE)