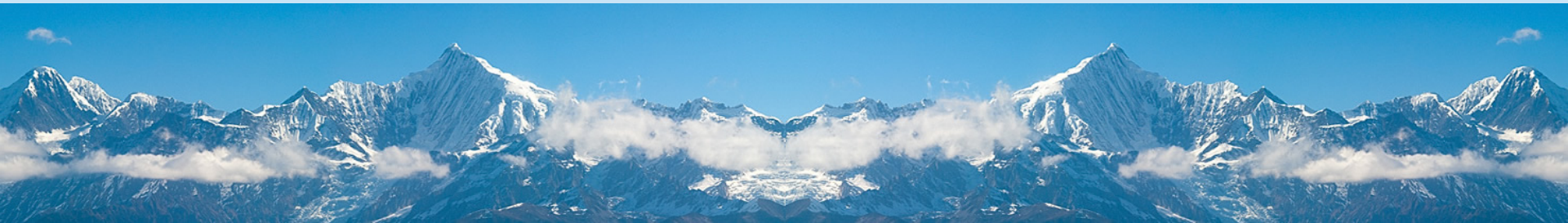
空间数据库2016秋季

空间查询语言2-1027

陈斌

北京大学地球与空间科学学院

gischen@pku.edu.cn



空间查询语言

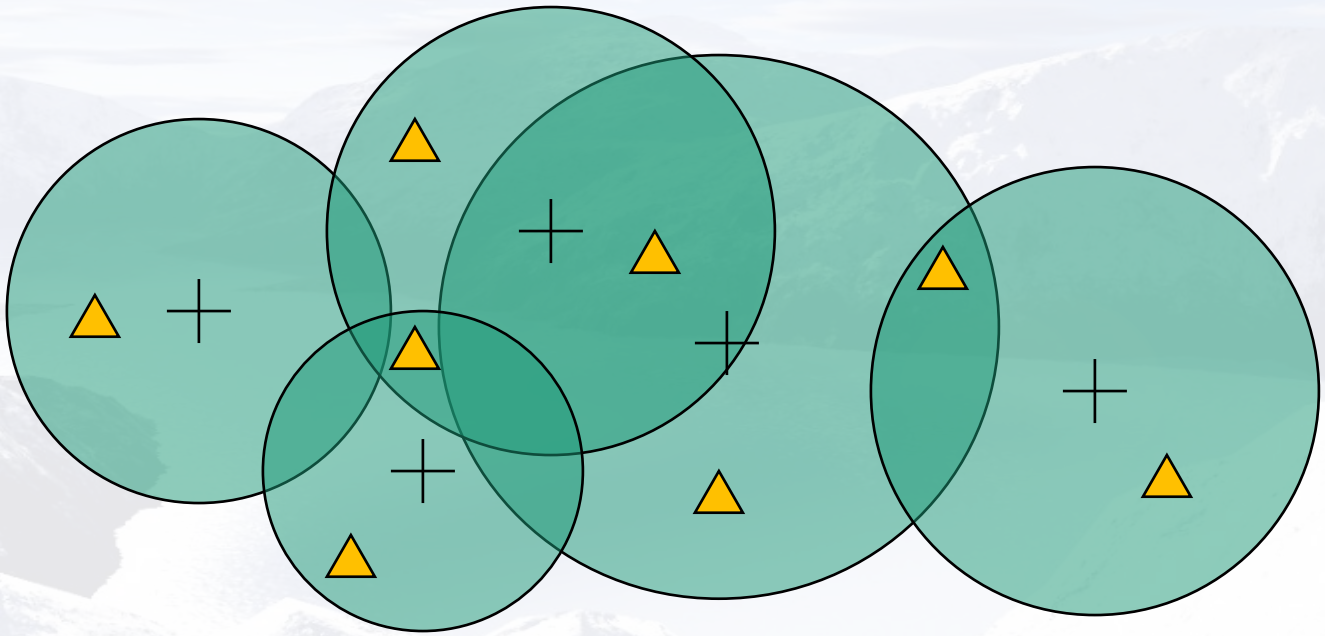
- 关系代数
- SQL语言
- SQL语言的空间扩展
- 空间查询
- 对象关系数据库及其空间扩展

SQL语言的空间扩展

- 直接的想法：全关系化的几何体
- 严格遵守1NF，将几何属性分解为无结构无嵌套的简单类型表示
- 利用SQL中表达式的能力来表示空间关系，进行空间运算
- 必要的话采用SQL过程机制来解决一些稍复杂的运算

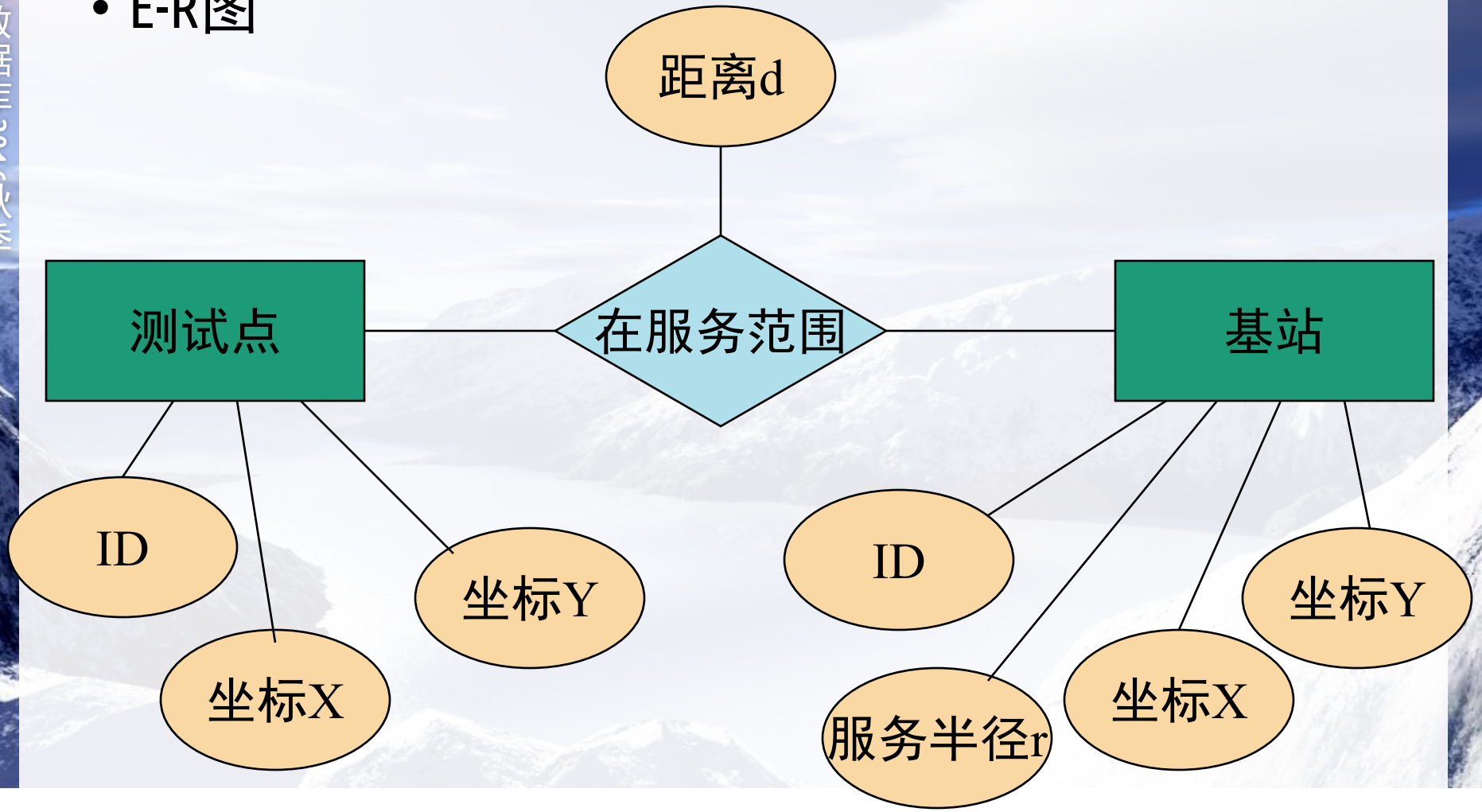
例子：移动通信

- 假想场景：对基站服务范围测试
  - 若干基站，具有坐标和理想服务范围（圆）
  - 若干测试点，具有坐标



例子：移动通信

- E-R图



例子：移动通信

- 关系模型（*为主码，#为外码）
  - station(*id, x, y, r)
  - monitor(*id, x, y)
  - in_range(#station_id, #monitor_id, distance)
- 基站插入
  - insert into station values (15, 23.4, 34.5, 500)
  - insert into in_range
    select s.id, m.id, sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))
    from station s, monitor m
    where s.id=15 and sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))<= s.r

例子：移动通信

- 测试点插入
  - insert into monitor values (6, 12.2, 11.3)
  - insert into in_range
    select s.id, m.id, sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))
    from station s, monitor m
    where m.id=6 and sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))<= s.r

例子：移动通信

- 基站删除
  - delete from station where id=15
  - delete from in_range where station_id=15
- 测试点删除
  - delete from monitor where id=6
  - delete from in_range where monitor_id=6

例子：移动通信

- 基站位置更新
  - update station set x=34.5 where id=15
  - delete from in_range where station_id=15
  - insert into in_range
    select s.id, m.id, sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))
    from station s, monitor m
    where s.id=15 and sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))<= s.r

例子：移动通信

- 测试点位置更新
  - update monitor set y=y+10 where id=6
  - delete from in_range where monitor_id=6
  - insert into in_range
    select s.id, m.id, sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))
    from station s, monitor m
    where m.id=6 and sqrt((s.x-m.x)*(s.x-m.x)+(s.y-m.y)*(s.y-m.y))<= s.r

例子：移动通信

- 查询每个基站包含测试点的数量
  - select station_id, count(*)
    from in_range
    group by station_id
- 查询服务范围内没有测试点的基站
  - select id from station
    except
    select distinct station_id from in_range

例子：移动通信

- 查询没有公共基站的测试点对
  - select m1.id, m2.id
    from monitor m1, monitor m2
    where not exists (
    select *
    from in_range i1, in_range i2
    where i1.monitor_id=m1.id and
    i2.monitor_id=m2.id and
    i1.station_id=i2.station_id)

例子：移动通信

- 查询每个基站服务区内距离最远的测试点
  - select station_id, monitor_id
    from in_range i1
    where distance >= all(
    select distance
    from in_range
    where station_id=i1.station_id)

例子分析

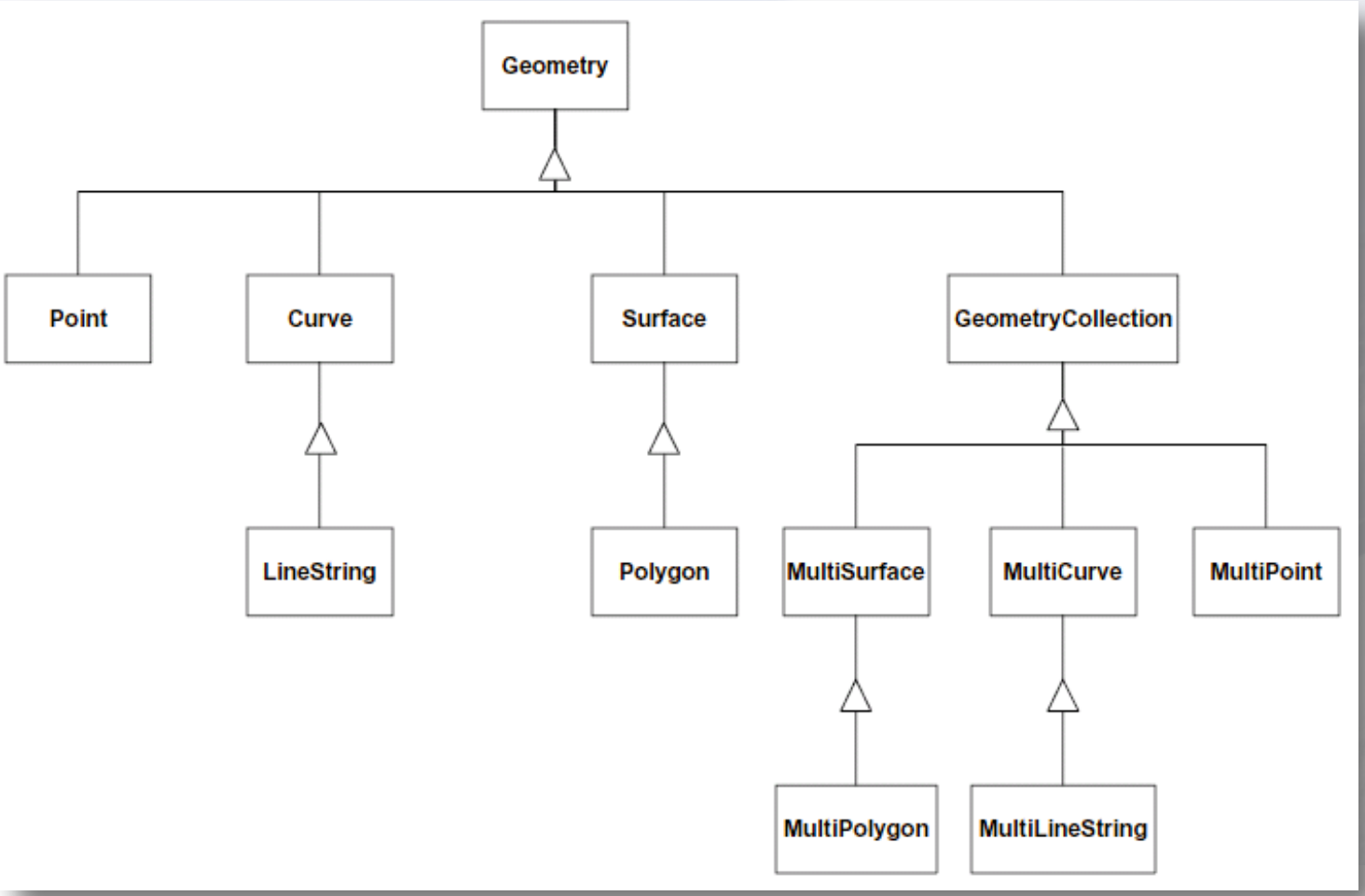
- 适用于点或者由点定义的定长结构
  - 如：圆(x,y,r)，正方形(x,y,l)，矩形(x,y,w,h)
  - 线段(x1,y1,x2,y2)
- 对于变长的结构，难度极大
  - 采用定长冗余结构：限制顶点个数
  - 采用外置表记录顶点，并设置外码参照
  - 破坏1NF原则，采用变长结构
  - 空间关系判定和空间运算极其复杂，实用价值不高

OpenGIS的SQL92实现规范

- 以SQL92标准实现OGM Geometry类型
  - 定义了Feature Table的存储方案
  - 定义了Geometry的存储方案
  - 定义了SRS信息的存储方案
  - 没有定义关于访问、维护、索引几何形的SQL函数
    - 这些函数在支持SQL92标准的数据库系统中不能标准地实现

SQL92 Geometry实现

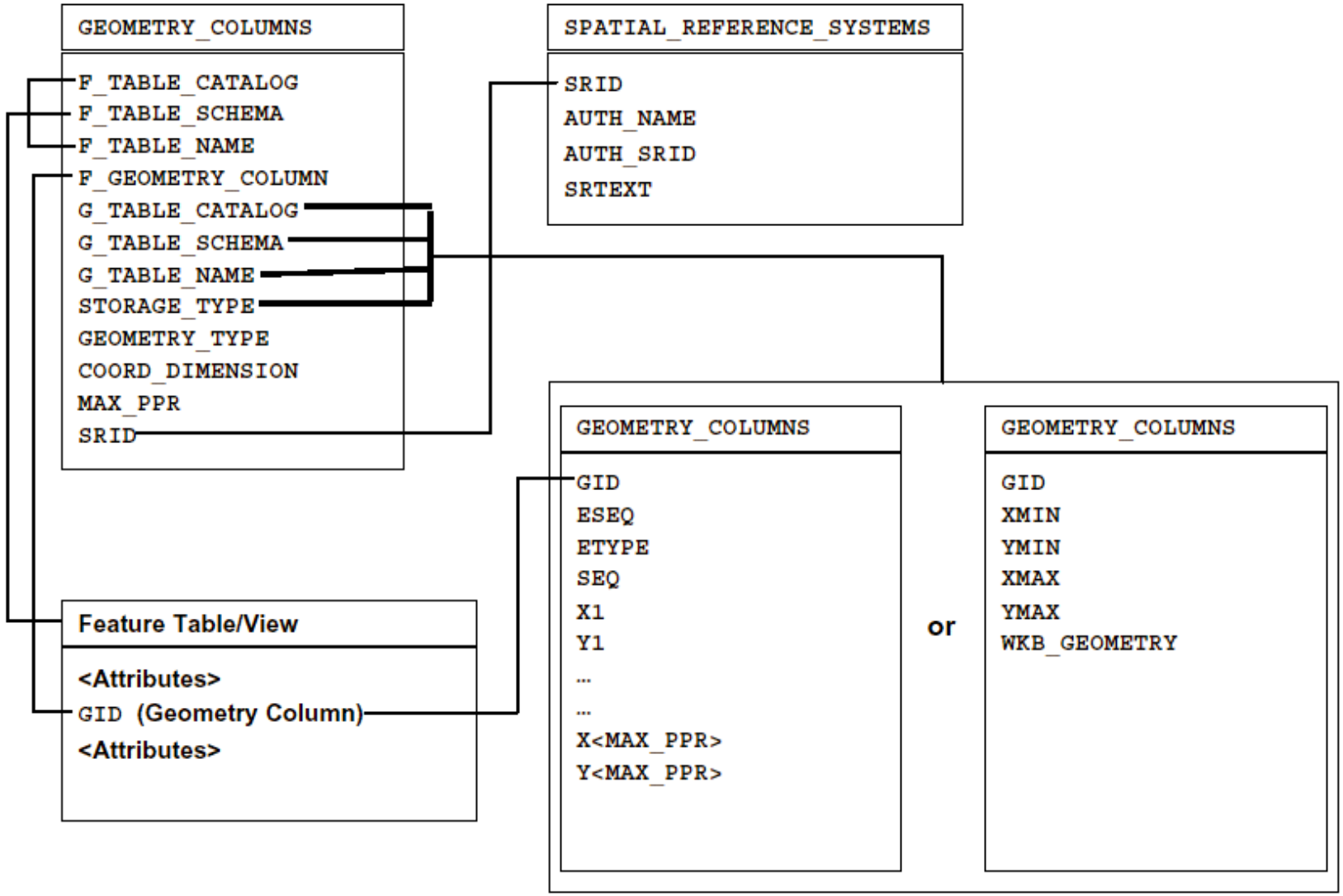
- SQL Geometry类层次图



SQL92实现-结构

- 支持OGIS简单Feature的数据库模式
  - 每个Feature表或者视图对应一个OGIS Feature类
  - Feature表中每一行对应一个Feature对象
  - 每个Feature都包含着若干几何属性，对应Feature表的几何类型字段
  - 每个几何类型字段都对应着一个特殊的Geometry表，表中存储了同一空间参照系下的若干Geometry对象
  - Feature对象和Geometry对象的对应关系是通过Feature表中的GID进行外码参照的，而Geometry表以GID作为主码

SQL92实现-模式图


GEOMETRY_COLUMNS
F_TABLE_CATALOG
F_TABLE_SCHEMA
F_TABLE_NAME
F_GEOMETRY_COLUMN
G_TABLE_CATALOG
G_TABLE_SCHEMA
G_TABLE_NAME
STORAGE_TYPE
GEOMETRY_TYPE
COORD_DIMENSION
MAX_PPR
SRID
SPATIAL_REFERENCE_SYSTEMS
SRID
AUTH_NAME
AUTH_SRID
SRTEXT
Feature Table/View
<Attributes>
GID (Geometry Column)
<Attributes>
GEOMETRY_COLUMNS
GID
ESEQ
ETYPE
SEQ
X1
Y1
…
…
X<MAX_PPR>
Y<MAX_PPR>
or
GEOMETRY_COLUMNS
GID
XMIN
YMIN
XMAX
YMAX
WKB_GEOMETRY

SQL92实现-结构

- 系统表GEOMETRY_COLUMNS
  - 存储数据库中Geometry字段的情况
  - 每一行指明空间数据库中的一个Geometry字段，包括
    - Geometry字段所在的Feature表名
    - 空间参照系统ID
    - 字段包含的Geometry的类型
    - 字段包含的Geometry的坐标维度
    - 字段的名称
    - Geometry的存储格式

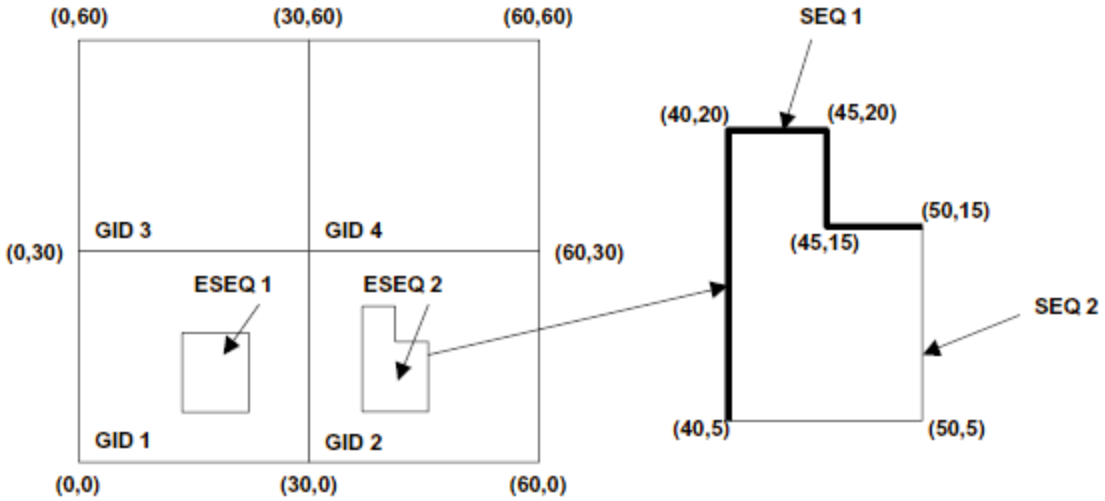
SQL92实现-结构

- 系统表SPATIAL_REFERENCE_SYSTEM
  - 存储数据库中空间参照系的信息
  - 每一行描述了一个空间参照系，信息包括
    - SRID：空间参照系的标识号
    - 空间参照系的通用名称和通用ID
    - 对空间参照系的标准文本描述，不同的客户端可以通过对标准描述的分析得到空间参照系的详细信息
- 系统表GEOMETRY
  - 记录了数据库中所有的Geometry对象

SQL92实现-结构

- Geometry对象存储格式：坐标序列存储
  - 采用SQL数据类型记录坐标值
  - 客户端应该根据查询的坐标值建立Geometry对象
  - 标准结构包括了：
    - 唯一标识Geometry对象的GID
    - 如果Geometry对象包含多个元素，ESEQ指明了元素的序列号
    - ETYPE指明元素的几何形类型
    - 一个元素可能需要多行来表示，SEQ指明行顺序
    - 若干个存储坐标对的字段

SQL92实现-结构

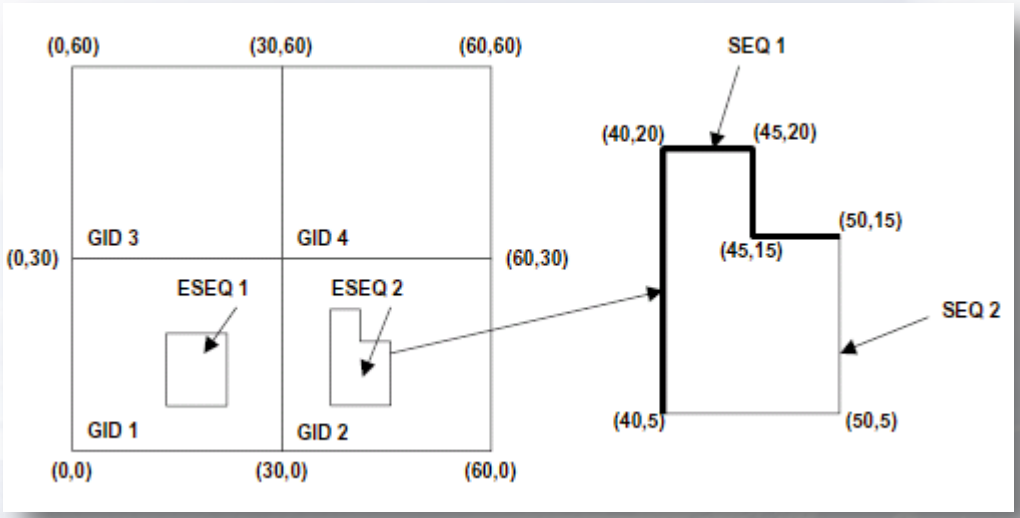


GID	ESEQ	ETYPE	SEQ	X0	Y0	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	1	3	1	0	0	0	30	30	30	30	0	0	0
1	2	3	1	10	10	10	20	20	20	20	10	10	10
2	1	3	1	30	0	30	30	60	30	60	0	30	0
2	2	3	1	40	5	40	20	45	20	45	15	50	15
2	2	3	2	50	15	50	5	40	5	Nil	Nil	Nil	Nil
3	1	3	1	0	30	0	60	30	60	30	30	0	30
4	1	3	1	30	30	30	60	60	60	60	30	30	30

SQL92实现-结构

- Geometry对象存储格式：二进制存储
  - 采用WKB（众所周知的二进制）格式存储Geometry对象
  - 客户端可以直接将数据包进行对象生成，也可以解开数据包访问其中的个别坐标
  - 标准结构包括了
    - 唯一标识Geometry对象的GID，并作为主码
    - Geometry对象的最小包围盒
    - Geometry对象数据的编码数据

SQL92实现-结构



GID	XMIN	YMIN	XMAX	YMAX	GEOMETRY
1	0	0	30	30	< WKBGeometry>
2	30	0	60	30	< WKBGeometry >
3	0	30	30	60	< WKBGeometry >
4	30	30	60	60	< WKBGeometry >

OGIS标准-SQL92+G

- Geometry字段元数据

```
CREATE TABLE GEOMETRY_COLUMNS (
  F_TABLE_CATALOG VARCHAR(256) NOT NULL,
  F_TABLE_SCHEMA VARCHAR(256) NOT NULL,
  F_TABLE_NAME VARCHAR(256) NOT NULL,
  F_GEOMETRY_COLUMN VARCHAR(256) NOT NULL,
  G_TABLE_CATALOG VARCHAR(256) NOT NULL,
  G_TABLE_SCHEMA VARCHAR(256) NOT NULL,
  G_TABLE_NAME VARCHAR(256) NOT NULL,
  STORAGE_TYPE INTEGER,
  GEOMETRY_TYPE INTEGER,
  COORD_DIMENSION INTEGER,
  MAX_PPR INTEGER, //max point per row
  SRID INTEGER REFERENCES SPATIAL_REF_SYS,
  CONSTRAINT GC_PK PRIMARY KEY
  (F_TABLE_CATALOG, F_TABLE_SCHEMA,
   F_TABLE_NAME, F_GEOMETRY_COLUMN))
```

0:坐标表示
1:二进制编码

0 = GEOMETRY
1 = POINT
2 = CURVE
3 = LINESTRING
4 = SURFACE
5 = POLYGON
6 = COLLECTION
7 = MULTIPOINT
8 = MULTICURVE
9 = MULTILINESTRING
10 = MULTISURFACE
11 = MULTIPOLYGON

OGIS标准-SQL92+G

- Feature表格式

```
CREATE TABLE <feature-name> (
<FID name> <FID type>,
<feature attributes> <other FID type> REFERENCES <other feature view>,
… (other FID based attributes for feature relations)
… (other attributes for feature)
<geometry attribute 1> <GID type>,
… (other geometric attributes for feature)
PRIMARY KEY <FID name>,
… (other geometric attributes foreign key statements)
FOREIGN KEY <geometric attribute 1> REFERENCES <geometry-table-name-1>,
FOREIGN KEY <FID relation name> REFERENCES <FEATURE table> <other FID name>,
… (other geometric attributes foreign key statements)
)
```

OGIS标准-SQL92+G

- Geometry表或者视图-SQL数类型

```
CREATE TABLE <table name> (
  GID NUMBER NOT NULL,
  ESEQ INTEGER NOT NULL,
  ETYPE INTEGER NOT NULL,
  SEQ INTEGER NOT NULL,
  X1 <ordinate type>,
  Y1 <ordinate type>,
... <repeated for each ordinate, repeated for each point>
  X<max_ppr> <ordinate type>,
  Y<max_ppr> <ordinate type>,
...,
  <attribute name> <attribute type>
  CONSTRAINT GID_PK PRIMARY KEY (GID, ESEQ, SEQ)
)
```

1 = Point
2 = LineString
3 = Polygon

OGIS标准-SQL92+G

- Geometry表或者视图-二进制编码

```
CREATE TABLE <table name> (
  GID NUMBER NOT NULL PRIMARY KEY,
  XMIN <ordinate type>,
  YMIN <ordinate type>,
  XMAX <ordinate type>,
  YMAX <ordinate type>,
  WKB_GEOMETRY VARBINARY,
  <attribute name> <attribute type>
)
```

OGIS标准-SQL92+G

- 空间参照系信息

```
CREATE TABLE SPATIAL_REF_SYS
(
   SRID INTEGER NOT NULL PRIMARY KEY,
   AUTH_NAME VARCHAR (256),
   AUTH_SRID INTEGER,
   SRTEXT VARCHAR (2048)
)
```

对关系数据模型进行空间扩展

- 对表示实体的关系进行空间扩展
  - 增加空间数据类型
  - 增加空间索引
- 对表示联系的关系进行空间扩展
  - 部分关系通过空间关系成为隐含的

对关系数据模型进行空间扩展

- 对关系运算进行空间扩展
  - 选择：基于空间关系谓词的过滤
  - 投影：不变
  - 连接：基于空间关系谓词的空间连接
  - 并、交、差
    - 基于空间操作Overlay的几何并、交、差

对象关系数据库和SQL3

- 对象关系模型扩展了关系模型
- 支持用户自定义数据类型
- ORDBMS提供机制实现UDT的无缝集成
  - SQL语言
  - 存储和索引
  - 一致性、完整性
  - 安全性
  - 事务支持
- ADT和ROWTYPE

对象关系数据库的空间扩展

- Oracle Spatial
- DB2 Spatial & Geodetic
- Informix Universal Server Datablade
- PostgreSQL
- PostGIS

OpenGIS的SQL92+G实现规范

- 抽象数据类型ADT(Abstract Data Type)
  - 可以在SQL系统中扩展数据类型
  - ADT类型可以用于字段定义
  - SQL函数可以将ADT类型作为参数，可以返回ADT类型的值
  - 一个ADT类型可以定义为另一个ADT类型的子类型，子类型可以出现在所有超类型能够出现的场合
- 可以采用ADT在SQL系统中实现OGIS几何对象模型

SQL92+Geometry实现

- OGIS标准不涉及Geometry是如何实现，包括
  - 类型定义的语法和功能
  - SQL函数定义的语法和功能
  - Geometry对象如何存储在数据库中
  - 用来实现几何对象模型的技术，如ADT
- OGIS标准提供如下的标准：
  - OpenGIS Geometry类型的命名和几何定义
  - OpenGIS SQL函数的命名、参数和几何定义

SQL92+Geometry实现

- Geometry的值和空间参照系
  - 几何对象的坐标值都对应特定的SRID，在空间操作中需要匹配SRID
    - 在生成和插入几何对象时，SRID需要和Feature表中定义的SRID对应
    - 在进行空间连接查询时，来自不同Feature表对应的SRID必须相容
  - 在Feature表定义中可以指定SRID的完整性规则

SQL92+Geometry实现

- SRID完整性约束示例

```
CREATE TABLE Countries (
  Name VARCHAR(200) NOT NULL PRIMARY KEY,
  Location Polygon NOT NULL,
  CONSTRAINT spatial_reference
  CHECK (SRID(Location) in
    (SELECT SRID from GEOMETRY_COLUMNS where
     F_TABLE_CATALOG = <catalog> and
     F_TABLE_SCHEMA = <schema> and
     F_TABLE_NAME =  'Countries'  and
     F_GEOMETRY_COLUMN =  'Location' ))
```

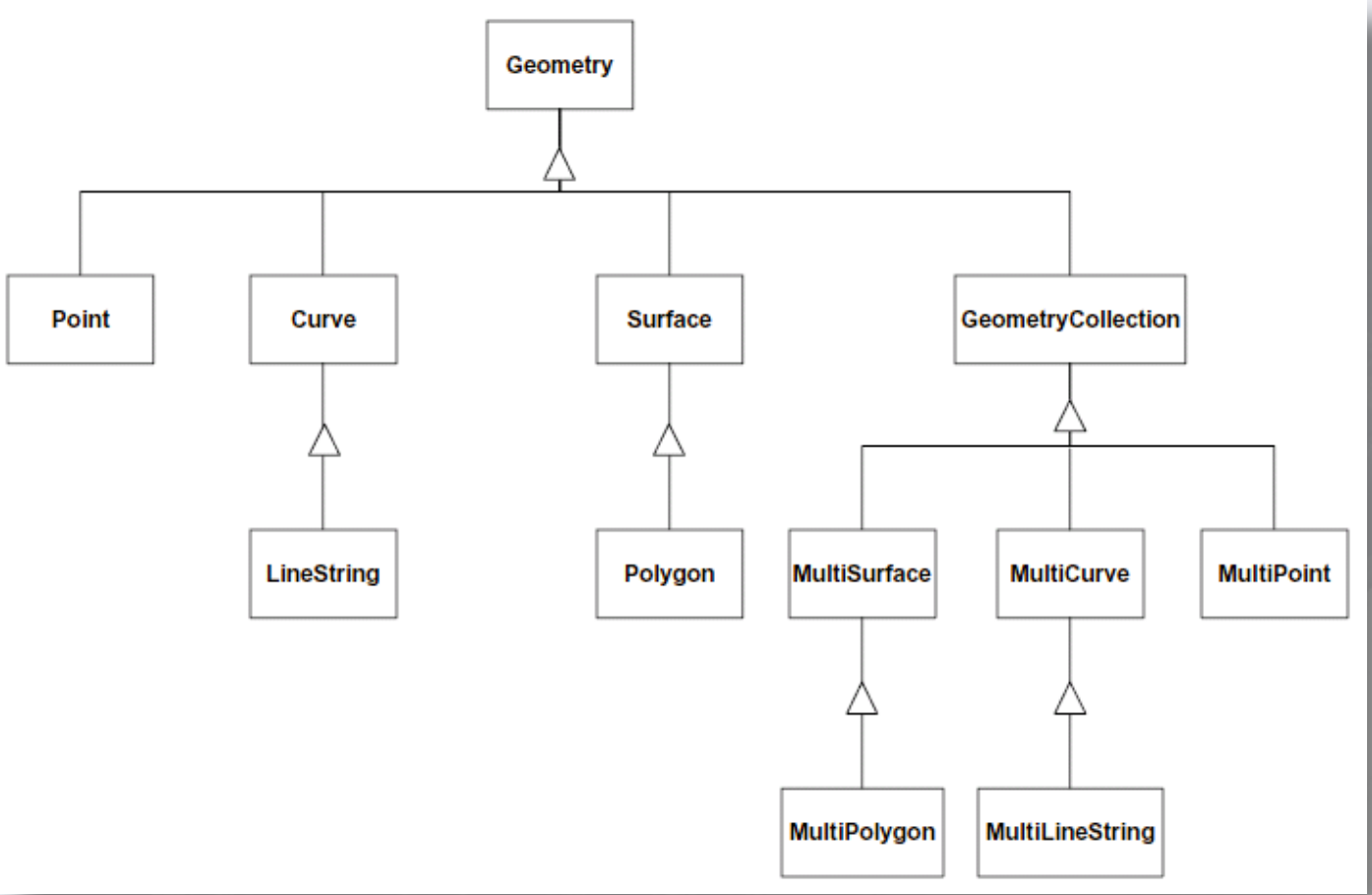
OGIS标准-SQL92+G

- 空间参照系信息

```
CREATE TABLE SPATIAL_REF_SYS
(
   SRID INTEGER NOT NULL PRIMARY KEY,
   AUTH_NAME VARCHAR (256),
   AUTH_SRID INTEGER,
   SRTEXT VARCHAR (2048)
)
```

OGIS标准-SQL92+G

- SQL Geometry类型



OGIS标准-SQL92+G

- SQL Geometry类型支持级别

支持层次	支持类型	可实例化
1	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection	Point, LineString, Polygon, GeomCollection
2	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection, MultiPoint, MultiCurve, MultiLineString, MultiSurface, MultiPolygon	Point, LineString, Polygon, MultiPoint, MultiLineString,MultiPolygon
3	Geometry, Point, Curve, LineString, Surface, Polygon, GeomCollection, MultiPoint , MultiCurve, MultiLineString, MultiSurface, MultiPolygon	Point, LineString, Polygon, GeomCollection, MultiPoint, MultiLineString, MultiPolygon

OGIS标准-SQL92+G

- Feature表格式

CREATE TABLE <feature-name> (
<FID name> <FID type>,
<feature attributes> <other FID type> REFERENCES <other feature view>,
… (other FID based attributes for feature relations)
… (other attributes for feature)
<geometry attribute 1> <Geometry type>,
… (other geometric attributes for feature)
PRIMARY KEY <FID name>,
FOREIGN KEY <FID relation name> REFERENCES <FEATURE table> <other FID name>
CONSTRAINT SRS_1 CHECK (SRID(<geometry attribute 1>) in (SELECT SRID from GEOMETRY_COLUMNS where F_TABLE_CATALOG = <catalog> and F_TABLE_SCHEMA = <schema> and F_TABLE_NAME = <feature-name> and F_GEOMETRY_COLUMN = <geometry attribute 1>))
... (spatial reference constraints for other geometric attributes))

OGIS标准-SQL92+G

- SQL中Geometry的文本表示

<Geometry Tagged Text> :=
<Point Tagged Text>
| <LineString Tagged Text>
| <Polygon Tagged Text>
| <MultiPoint Tagged Text>
| <MultiLineString Tagged Text>
| <MultiPolygon Tagged Text>
| <GeometryCollection Tagged Text>
<Point Tagged Text> :=
POINT <Point Text>
<LineString Tagged Text> :=
LINESTRING <LineString Text>

<Polygon Tagged Text> :=
POLYGON <Polygon Text>
<MultiPoint Tagged Text> :=
MULTIPOINT <Multipoint Text>
<MultiLineString Tagged Text> :=
MULTILINESTRING <MultiLineString Text>
<MultiPolygon Tagged Text> :=
MULTIPOLYGON <MultiPolygon Text>
<GeometryCollection Tagged Text> :=
GEOMETRYCOLLECTION <GeometryCollection Text>
<Point Text> := EMPTY | (<Point>)
<Point> := <x> <y>
<x> := double precision literal
<y> := double precision literal

OGIS标准-SQL92+G

- SQL中Geometry的文本表示

```
<LineString Text> := EMPTY
| ( <Point > {, <Point > }* )
<Polygon Text> := EMPTY
| ( <LineString Text > {, < LineString Text > }*)
<Multipoint Text> := EMPTY
| ( <Point Text > {, <Point Text > }* )
<MultiLineString Text> := EMPTY
| ( <LineString Text > {, < LineString Text > }* )
<MultiPolygon Text> := EMPTY
| ( < Polygon Text > {, < Polygon Text > }* )
<GeometryCollection Text> := EMPTY
| ( <Geometry Tagged Text> {, <Geometry Tagged Text> }* )
```

OGIS标准-SQL92+G

- SQL中Geometry的文本表示：示例

Geometry Type	SQL Text Literal Representation	Comment
Point	`'POINT (10 10)'`	a `Point`
LineString	`'LINESTRING ( 10 10, 20 20, 30 40)'`	a `LineString` with 3 points
Polygon	`'POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))'`	a `Polygon` with 1 exterior ring and 0 interior rings
Multipoint	`'MULTIPOINT (10 10, 20 20)'`	a `MultiPoint` with 2 point
MultiLineString	`'MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))'`	a `MultiLineString` with 2 linestrings
MultiPolygon	`'MULTIPOLYGON ( ((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60 ) ))'`	a `MultiPolygon` with 2 polygons
GeomCollection	`'GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))'`	a `GeometryCollection` consisting of 2 `Point` values and a `LineString` value

OGIS标准-SQL92+G

- 从文本形式构造几何对象的SQL函数

GeomFromText(geometryTaggedText String,SRID Integer) : Geometry
PointFromText (pointTaggedText String, SRID Integer): Point
LineFromText(lineStringTaggedText String,SRID Integer) : LineString
PolyFromText(polygonTaggedText String,SRID Integer): Polygon
MPointFromText (multiPointTaggedText String,SRID Integer): MultiPoint
MLineFromText (multiLineStringTaggedText String,SRID Integer): MultiLineString
MPolyFromText(multiPolygonTaggedText String,SRID Integer): MultiPolygon
GeomCollFromTxt(geometryCollectionTaggedText String,SRID Integer): GeomCollection

OGIS标准-SQL92+G

- 从MultiLineString生成Polygon

BdPolyFromText(multiLineStringTaggedText String,SRID Integer): Polygon
Construct a Polygon given an arbitrary collection of closed linestrings as a MultiLineString text representation.

BdMPolyFromText(multiLineStringTaggedText String,SRID Integer): MultiPolygon
Construct a MultiPolygon given an arbitrary collection of closed linestrings as a MultiLineString text representation.

OGIS标准-SQL92+G

- 从文本形式构造几何对象：例

```
INSERT INTO Countries (Name, Location)
VALUES ( 'Kenya' ,
    PolygonFromText(
       'POLYGON ((x y, x y, x y, ..., x y))' ,
      14))
```

OGIS标准-SQL92+G

- 从二进制形式构造几何对象的SQL函数
  - 同从文本形式构造，将FromText改成FromWKB即可

OGIS标准-SQL92+G

- 获取几何对象的文本形式的SQL函数
  - AsText (g Geometry) : String
  - 例：
    - SELECT Name, AsText(Location) FROM Countries WHERE Name LIKE 'K%'

OGIS标准-SQL92+G

- 获取几何对象的二进制形式的SQL函数
  - AsBinary (g Geometry) : Binary
  - 例：
    - SELECT Name, AsBinary(Location) FROM Countries WHERE Name LIKE 'K%'

OGIS标准-SQL92+G

- 在各种几何形类上定义的SQL函数
  - Geometry, GeomCollection
  - Point
  - Curve, LineString, MultiCurve
  - Surface, Polygon, MultiSurface

OGIS标准-SQL92+G

- 测试空间关系的SQL函数
  - Equals(g1 Geometry,g2 Geometry) : Integer
  - Disjoint(g1 Geometry, g2 Geometry) : Integer
  - Touches(g1 Geometry, g2 Geometry) : Integer
  - Within(g1 Geometry, g2 Geometry) : Integer
  - Overlaps(g1 Geometry, g2 Geometry) : Integer
  - Crosses(g1 Geometry, g2 Geometry) : Integer
  - Intersects(g1 Geometry, g2 Geometry) :Integer
  - Contains(g1 Geometry, g2 Geometry) : Integer
  - Relate(g1 Geometry, g2 Geometry, patternMatrix String) : Integer

OGIS标准-SQL92+G

- SQL查询示例
- 查询指定区域内的所有学校
  - select school.name
  - from school
  - where Within(school.location, PolygonFromWKB(:wkb, :srid))=1

OGIS标准-SQL92+G

- 距离函数
- Distance(g1 Geometry, g2 Geometry) : Double Precision
- 示例：查找距离指定点2000公里以内的机场
  - select airport.name
  - from airport
  - where Distance(PointFromText(:pT, :srid), airport.location)<2000

OGIS标准-SQL92+G

- 空间操作的SQL函数
- Intersection (g1 Geometry, g2 Geometry) : Geometry
- Difference(g1 Geometry, g2 Geometry) : Geometry
- Union(g1 Geometry, g2 Geometry) : Geometry
- SymDifference((g1 Geometry, g2 Geometry) : Geometry
- Buffer(g1 Geometry, d Double Precision) : Geometry
- ConvexHull (g1 Geometry) : Geometry

OGIS标准-SQL92+G

- 空间操作的SQL函数
- 示例：求指定坐落在河流两岸200米以内的所有工厂
  - select factory.name
  - from factory
  - where Within(factory.block, Buffer(LineStringFromText(:river, :srid), 200))=1

讨论组

- 空间数据库产品的查询语言特性讨论
  - Oracle Spatial
  - ArcGIS Geodatabase
  - PostGIS
  - Spatialite
  - MySQL Spatial
  - SQL Server
- 空间扩展的语言成分与功能
- 用空间查询语言来重做基站例子（p4～13）
- 对OpenGIS标准的支持程度
- 分为6个小组报告

参考文献

- [TP311.13/261]空间数据库 = Spatial databases a tour (美) Shashi Shekhar, Sanjay Chawla著 谢昆青 ... 等译 北京 机械工业出版社 2004
- OpenGIS Simple Features Specification for SQL, OGC, www.opengis.org
- The OpenGIS Abstract Specification: Feature, OGC, www.opengis.org