



Das UTM-System teilt die besiedelten Zonen der Erde in vertikale Streifen von 6 Längengraden bzw. maximal 800 km Breite auf, um diese jeweils einzeln auf einem kartesischen Koordinatensystem abzubilden. Diese Streifen (Geometriefelder) werden im WKT-Format abgespeichert. Das WKT-Format eines Geometriefeldes ermöglicht den Datenaustausch im ASCII-Format.

(Quelle: wikipedia.de)

GIS-Datentypen

1.	POINT speichert einen Punkt. z.B.: <code>POINT(x₁ y₁)</code>
2.	MULTIPOINT speichert eine Liste von Punkten. z.B.: <code>MULTIPOINT(x₁ y₁, x₂ y₂, x₃ y₃, . . . , x_n y_n)</code>
3.	LINESTRING speichert einen Linienzug. z.B.: <code>LINESTRING(x₁ y₁, x₂ y₂, x₃ y₃, . . . , x_n y_n)</code>
4.	MULTILINESTRING speichert mehrere Linienzüge. z.B.: <code>MULTILINESTRING((x₁ y₁, x₂ y₂), (x₁ y₁, x₂ y₂, x₃ y₃))</code>
5.	POLYGON speichert ein geschlossenes Polygon (Anfangspunkt = Endpunkt) und kann ggf. auch Löcher enthalten. z.B.: <code>POLYGON((x₁ y₁, x₂ y₂, x₃ y₃, x₁ y₁))</code> z.B.: <code>POLYGON((x₁ y₁, x₂ y₂, x₃ y₃, x₁ y₁), (x₁ y₁, x₂ y₂, x₁ y₁))</code>
6.	MULTIPOLYGON speichert mehrere geschlossene Polygone (ggf. auch mit Löchern). z.B.: <code>MULTIPOLYGON(((P₁), (L₁₁)), ((P₂), (L₂₁), (L₂₂), (L₃₂)))</code>
7.	GEOMETRYCOLLECTION speichert eine Liste von (beliebigen) Geometrieobjekten. z.B.: <code>GEOMETRYCOLLECTION(POINT(x₁ y₁), LINESTRING(x₁ y₁, x₂ y₂), . . .)</code>
8.	GEOMETRY kann jeden der vorherigen sieben Datentypen aufnehmen.

Indizieren von geometrischen Daten

INDEX	<pre>CREATE TABLE name (col1 Typ , col2 GeoDatentyp NOT NULL , SPATIAL INDEX(col2) Engine = MyISAM ;</pre>	<p>In MySQL ist es möglich, auch für raumbezogene Daten einen INDEX zu erzeugen, der die Abfragegeschwindigkeit erhöhen kann. Momentan (Vers 5.0.xx) werden InnoDB-Tabellen aber nicht unterstützt.</p>
--------------	---	---

Konvertierungs-Funktionen

Syntax	Beschreibung	Syntax	Beschreibung
<code>ASTEXT(geom)</code>	Liefert ein Geo-Objekt als Well-Known Text.	<code>GEOMFROMTEXT(text)</code>	Erzeugt ein Geo-Objekt aus einem Text.
<code>ASBINARY(geom)</code>	Liefert ein Geo-Objekt als Well-Known Binary.	<code>GEOMFROMWKB(binary)</code>	Erzeugt ein Geo-Objekt aus binären Daten.

Allgemeine Geometrie-Funktionen (für alle GIS-Typen)

Syntax	Beschreibung
DIMENSION (geom)	Liefert die Dimension eines Objektes. Mögliche Werte sind: -1 für leeres Obj. / 0 für Punkt / 1 für Linie / 2 für Polygon.
ENVELOPE (geom)	Liefert die Bounding-Box des Geo-Objektes als POLYGON.
GEOMETRYTYPE (geom)	Liefert den Typ des Geo-Objektes als Zeichenkette.

POINT-Funktionen

Syntax	Beschreibung	Syntax	Beschreibung
X (pt)	Liefert die X-Koordinate.	Y (pt)	Liefert die Y-Koordinate.

LINESTRING-Funktionen (teilweise auch für MULTILINE)

Syntax	Beschreibung	Syntax	Beschreibung
ENDPOINT (ls)	Liefert den letzten Punkt.	STARTPOINT (ls)	Liefert den 1. Punkt.
POINTN (ls, n)	Liefert den n-ten Punkt.	NUMPOINTS (ls)	Liefert die Punktzahl.
GLENGTH (ls)	Liefert die Linienlänge als Fließkommazahl.	ISCLOSED (ls)	Liefert 1 wenn Startpunkt = Endpunkt / sonst 0.

POLYGON-Funktionen (teilweise auch für MULTIPOLYGON)

Syntax	Beschreibung
AREA (p)	Liefert die Fläche des Polygons als Fließkommazahl.
EXTERIORRING (p)	Liefert den äußeren Ring des Polygons als LINESTRING-Objekt.
INTERIORRING (p, n)	Liefert den inneren Ring an der Stelle n als LINESTRING-Objekt.
NUMINTERIORRINGS (p)	Liefert die Anzahl der inneren Ringe (Löcher).

Analyse-Funktionen

Syntax	Beschreibung
[MBR] CONTAINS (g1, g2)	Liefert TRUE wenn g2 vollständig in g1 enthalten ist.
[MBR] WITHIN (g1, g2)	Liefert TRUE wenn g1 vollständig in g2 enthalten ist .
[MBR] EQUAL (g1, g2)	Liefert TRUE wenn g1 und g2 übereinstimmen.
[MBR] INTERSECTS (g1, g2)	Liefert TRUE wenn sich g1 und g2 schneiden.
[MBR] OVERLAPS (g1, g2)	Liefert TRUE wenn sich g1 und g2 überlappen.
[MBR] TOUCHES (g1, g2)	Liefert TRUE wenn sich g1 und g2 berühren.
[MBR] DISJOINT (g1, g2)	Liefert TRUE wenn g1 und g2 sich nicht überlappen oder berühren.

Hinweis : Jede Analysefunktion existiert in zwei Varianten. Die schnellere Variante berücksichtigt nur die Bounding-Box der Geometrieobjekte. MySQL kennt formal beide Varianten. Momentan (Vers 5.0.xx) sind intern beide identisch und berücksichtigen nur die Bounding-Box.