이번 프로젝트의 목표는 6개이상의 국가, 임의의 강을 포함한 5개 이상의 강으로 8가지의 관계를 구현하는 것이다.

먼저, 국가의 좌표는 다음과 같다.

italiana

37.943925, 12.063867, 36.578963, 15.065327, 37.849184, 15.991491, 39.071268, 17.243529, 39.812993, 16.523178, 40.481604, 17.003412, 39.799817, 18.409810, 40.324876, 18.495566, 41.811806, 16.283062, 42.308419, 14.533640, 43.576438, 13.641778, 45.302897, 12.458345, 45.651658, 13.521719, 47.036816, 12.475496, 45.890925, 6.867055, 43.731008, 7.552106, 43.996819, 9.872228, 39.851941, 8.133797,37.943925, 12.063867

France

39.851941, 8.133797, 49.867106, -1.895244, 48.216819, -5.420298, 43.494538, -1.943203, 42.441763, 3.236468, 43.581456, 7.600821, 48.930793, 8.200320, 51.148059, 2.397170, 39.851941, 8.133797

spain

43.040548, -9.137192, 37.235147, -9.209132, 35.906350, -5.564178, 38.559683, 0.238973, 42.370938, 3.332388, 43.650899, -2.135043, 43.040548, -9.137192

England

58.949794, -5.217394, 54.191357, -10.458454, 51.459448, -10.602044, 49.634879, -5.576370, 51.055070, 1.423676, 52.933519, 1.926243, 59.005296, -2.309682, 58.949794, -5.217394

swiss

47.494465, 6.831210, 46.140421, 5.812534, 45.904640, 7.814759, 45.831262, 8.962233, 46.237691, 10.168251, 46.841732, 10.578063, 47.557717, 9.559387, 47.731267, 8.119191, 47.494465, 6.831210

germany

54.965510, 8.579042, 53.631837, 6.857743, 53.631837, 6.857743, 49.495793, 6.519127, 50.904536, 14.984534, 54.180403, 14.222647, 54.884433, 13.347889, 54.591200, 11.118665, 54.868198, 9.651327, 54.965510, 8.579042

Austria

47.582771, 9.476611, 47.043727, 9.504028, 47.006348, 12.108577, 46.310128, 14.548628, 47.006348, 16.330688, 47.932959, 17.098344, 48.734882, 16.933846, 49.041341, 15.042121, 48.644389,

14.055134, 47.638221, 12.848817, 47.638221, 9.613693, 47.582771, 9.476611

강의 좌표는 다음과 같다.

lucerne river

46.987882, 8.525983, 45.500296, 10.655721, 43.736551, 12.123969, 43.144292, 13.801966

san river

48.948709, 2.305865, 48.985683, 2.083492, 49.112972, 1.453272, 49.442399, 0.217108

dordogne river

43.764270, 1.334080, 44.575004, -0.099210, 44.851722, -0.563378, 45.276961, -0.73622

elbe river

53.512585, 10.042253, 53.789327, 9.391313, 53.881711, 8.916155

make river(임의의 강)

46.693735, 14.754785, 50.725504, 8.117755, 44.051153, 0.696824, 42.837034, -5.469865

make2 river(임의의 강, cover 관계를 충족하기 위함.)

47.043727, 9.504028, 47.030000, 11.000000, 47.006348, 12.108577

make3 river(meet 관계를 위한 임의의 강

47.582771, 9.476611,45.582771, 9.476611)

각 테이블의 스키마는 다음과 같다.

Countrys 테이블

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SID	NUMBER	No	(null)	1	(null)
2	CNAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	CSHAPE	SDO_GEOMETRY	Yes	(null)	3	(null)

rivers 테이블

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RID	NUMBER	No	(null)	1	(null)
2	RNAME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	RSHAPE	SDO_GEOMETRY	Yes	(null)	3	(null)

DESC로 확인

sql이 익숙치 않아 설치부터 약간의 고생이 있었으나, 8가지 관계가 모두 잘 재현됨을 확인할수 있었다.

disjoint

```
/*disjoint */
SELECT cname2
FROM (
SELECT c1.cname cname1, c2.cname cname2, SDO_GEOM.RELATE(c1.cshape,'de
FROM countrys c1, countrys c2
)
WHERE relationship='DISJOINT';
/* contains */
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 28(0.007초)

CN...
spain
england
germany
england
austria
italiana
england
swiss
germany
austria
italiana
France
spain

contains

```
/* contains */
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rs
FROM countrys c1, rivers r2
)
WHERE relationship='CONTAINS';
```

스크립트 출력 x 질의 결과 x

SQL | 인출된 모든 행: 1(0.005초)

	CNAME	RNAME
1	France	dordogne river

inside

```
/*inside*/
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(r2.rshape,'determine',c1.cs
FROM countrys c1, rivers r2
)
WHERE relationship='INSIDE';
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.006초)

	CNAME	RNAME
1	France	dordogne river

equal

```
SELECT rname1
FROM (
SELECT r1.rname rname1, r2.rname rname2, SDO_GEOM.RELATE(r1.rshape,'c
FROM rivers r1, rivers r2
)
WHERE relationship='EQUAL';
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 6(0.005초)

	RNAME1
1	lucerne river
2	san river
3	dordogne river
4	elbe river
5	make river
6	make2 river

meet

```
/* meet */
SELECT *
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rs
FROM countrys c1, rivers r2
)
WHERE relationship='TOUCH';
/* covers */
SELECT cname, rname
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.007초)

	CNAME	RNAME	RELATIONSHIP
1	austria	make3 river	TOUCH

covers

```
/* covers */
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.r
FROM countrys c1, rivers r2
)
WHERE relationship='COVERS';
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.008초)

	CNAME	RNAME
1	austria	make2 river

coveredby

```
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(r2.rshape,'determine',c1.cs
FROM countrys c1, rivers r2
)
WHERE relationship='COVEREDBY';

/*overlapbdydisjoint */
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.006초)

	CNAME	RNAME
1	austria	make2 river

overlap

```
/*overlapbdydisjoint */
SELECT cname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rs
FROM countrys c1, rivers r2
)
WHERE relationship='OVERLAPBDYDISJOINT' and rname='make river';


/* 모든 관계 보기 */
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 3(0.003초)

	CNAME
1	France
2	spain
3	austria

sql문

```
CREATE TABLE countrys(
 sid NUMBER PRIMARY KEY,
 cname VARCHAR2(20),
 cshape SDO_GEOMETRY
);

CREATE TABLE rivers(
 rid NUMBER PRIMARY KEY,
 rname VARCHAR2(20),
 rshape SDO_GEOMETRY
);



INSERT INTO countrys VALUES(
 1,
 'italiana',
 SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
  SDO_ORDINATE_ARRAY(37.943925, 12.063867, 36.578963, 15.065327, 37.849184, 15.991491, 39.071268, 17.243529, 39.812993, 16.523178, 40.481604, 17.003412, 39.799817, 18.409810, 40.324876, 18.495566, 41.811806, 16.283062, 42.308419, 14.533640, 43.576438, 13.641778, 45.302897, 12.458345, 45.651658, 13.521719, 47.036816, 12.475496, 45.890925, 6.867055, 43.731008, 7.552106, 43.996819, 9.872228, 39.851941, 8.133797,37.943925, 12.063867
  )
 )
);
```

```
INSERT INTO countrys VALUES(
 2,
 'France',
 SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(39.851941, 8.133797, 49.867106, -1.895244, 48.216819, -5.420298, 43.494538, -1.943203, 42.441763, 3.236468, 43.581456, 7.600821, 48.930793, 8.200320, 51.148059, 2.397170, 39.851941, 8.133797

  )
 )
);

INSERT INTO countrys VALUES(
 3,
 'spain',
 SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(43.040548, -9.137192, 37.235147, -9.209132, 35.906350, -5.564178, 38.559683, 0.238973, 42.370938, 3.332388, 43.650899, -2.135043, 43.040548, -9.137192

  )
 )
);
```

```
INSERT INTO countrys VALUES(
 4,
 'england',
 SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(58.949794, -5.217394, 54.191357, -10.458454, 51.459448, -10.602044, 49.634879, -5.576370,
51.055070, 1.423676, 52.933519, 1.926243, 59.005296, -2.309682, 58.949794, -5.217394



  )
 )
);



INSERT INTO countrys VALUES(
 5,
 'swiss',
 SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
```

```
      SDO_ORDINATE_ARRAY(47.494465, 6.831210, 46.140421, 5.812534, 45.904640, 7.814759, 45.831262, 8.962233, 46.237691, 10.168251, 46.841732, 10.578063, 47.557717, 9.559387, 47.731267, 8.119191, 47.494465, 6.831210



    )
  )
);

INSERT INTO countrys VALUES(
  6,
  'germany',
  SDO_GEOMETRY(
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1),
      SDO_ORDINATE_ARRAY(54.965510, 8.579042, 53.631837, 6.857743, 53.631837, 6.857743, 49.495793, 6.519127, 50.904536, 14.984534, 54.180403, 14.222647, 54.884433, 13.347889, 54.591200, 11.118665, 54.868198, 9.651327, 54.965510, 8.579042



    )
  )
);

INSERT INTO countrys VALUES(
  7,
  'austria',
  SDO_GEOMETRY(
    2003,
    NULL,
```

```
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1),
      SDO_ORDINATE_ARRAY(47.582771, 9.476611, 47.043727, 9.504028, 47.006348, 12.108577, 46.310128, 14.548628, 47.006348, 16.330688, 47.932959, 17.098344, 48.734882, 16.933846, 49.041341, 15.042121, 48.644389, 14.055134, 47.638221, 12.848817, 47.638221, 9.613693, 47.582771, 9.476611
    )
  )
);

INSERT INTO rivers VALUES(
  1,
  'lucerne river',
  SDO_GEOMETRY(
    2002,
    NULL,
```

```
    NULL,
    SDO_ELEM_INFO_ARRAY(1,2,1),
    SDO_ORDINATE_ARRAY(46.987882, 8.525983, 45.500296, 10.655721, 43.736551, 12.123969, 43.144292, 13.801966)
  )
);



INSERT INTO rivers VALUES(
  2,
  'san river',
  SDO_GEOMETRY(
    2002,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,2,1),
    SDO_ORDINATE_ARRAY(48.948709, 2.305865, 48.985683, 2.083492, 49.112972, 1.453272, 49.442399, 0.217108)
  )
);

INSERT INTO rivers VALUES(
  3,
  'dordogne river',
  SDO_GEOMETRY(
    2002,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,2,1),
    SDO_ORDINATE_ARRAY(43.764270, 1.334080, 44.575004, -0.099210, 44.851722, -0.563378, 45.276961, -0.736227)
  )
```

```
);

INSERT INTO rivers VALUES(
 4,
 'elbe river',
 SDO_GEOMETRY(
  2002,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,2,1),
  SDO_ORDINATE_ARRAY(53.512585, 10.042253, 53.789327, 9.391313, 53.881711, 8.916155)
 )
);

INSERT INTO rivers VALUES(
 5,
 'make river',
 SDO_GEOMETRY(
  2002,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,2,1),
  SDO_ORDINATE_ARRAY(46.693735, 14.754785, 50.725504, 8.117755, 44.051153, 0.696824, 42.837034, -5.469865)
 )
);

INSERT INTO rivers VALUES(
```

```
  6,
  'make2 river',
  SDO_GEOMETRY(
    2002,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,2,1),
    SDO_ORDINATE_ARRAY(47.043727, 9.504028, 47.030000, 11.000000, 47.006348, 12.108577)
  )
);

INSERT INTO rivers VALUES(
  7,
  'make3 river',
  SDO_GEOMETRY(
    2002,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,2,1),
    SDO_ORDINATE_ARRAY(47.582771, 9.476611,45.582771, 9.476611)
  )
);

/*disjoint */
SELECT cname2
FROM (
```

```
SELECT c1.cname cname1, c2.cname cname2, SDO_GEOM.RELATE(c1.cshape,'determine',c2.cshape,0.001) relationship
FROM countrys c1, countrys c2
)
WHERE relationship='DISJOINT';
/* contains */
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rshape,0.001) relationship
FROM countrys c1, rivers r2
)
WHERE relationship='CONTAINS';
/*inside*/
SELECT cname, rname
FROM (
SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(r2.rshape,'determine',c1.cshape,0.001) relationship
FROM countrys c1, rivers r2
)
WHERE relationship='INSIDE';
/* equal */
SELECT *
FROM (
SELECT c1.cname cname1, c2.cname cname2, SDO_GEOM.RELATE(c1.cshape,'determine',c2.cshape,0.001) relationship
FROM countrys c1, countrys c2
)
WHERE relationship='EQUAL';
SELECT rname1
FROM (
SELECT r1.rname rname1, r2.rname rname2, SDO_GEOM.RELATE(r1.rshape,'determine',r2.rshape,0.001) relationship
FROM rivers r1, rivers r2
)
```

```
WHERE relationship='EQUAL';

/* meet */

SELECT *

FROM (

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rshape,0.001) relationship

FROM countrys c1, rivers r2

)

WHERE relationship='TOUCH';

/* covers */

SELECT cname, rname

FROM (

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rshape,0.001) relationship

FROM countrys c1, rivers r2

)

WHERE relationship='COVERS';

/* coverdby */

SELECT cname, rname

FROM (

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(r2.rshape,'determine',c1.cshape,0.001) relationship

FROM countrys c1, rivers r2

)

WHERE relationship='COVEREDBY';


/*overlapbdydisjoint */

SELECT cname

FROM (

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rshape,0.001) relationship

FROM countrys c1, rivers r2

)

WHERE relationship='OVERLAPBDYDISJOINT' and rname='make river';
```

```
/* 모든 관계 보기 */
SELECT r1.rname, r2.rname, SDO_GEOM.RELATE(r1.rshape,'determine',r2.rshape,0.001) relationship
FROM rivers r1, rivers r2;

SELECT c1.cname, c2.cname, SDO_GEOM.RELATE(c1.cshape,'determine',c2.cshape,0.001) relationship
FROM countrys c1, countrys c2;

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(c1.cshape,'determine',r2.rshape,0.001) relationship
FROM countrys c1, rivers r2;

SELECT c1.cname, r2.rname, SDO_GEOM.RELATE(r2.rshape,'determine',c1.cshape,0.001) relationship
FROM countrys c1, rivers r2;
```