SI 664

Meeting 5

2 Oct 2018

Anthony Whyte
arwhyte@umich.edu
twitter: @arwhyte
University of Michigan

preliminaries

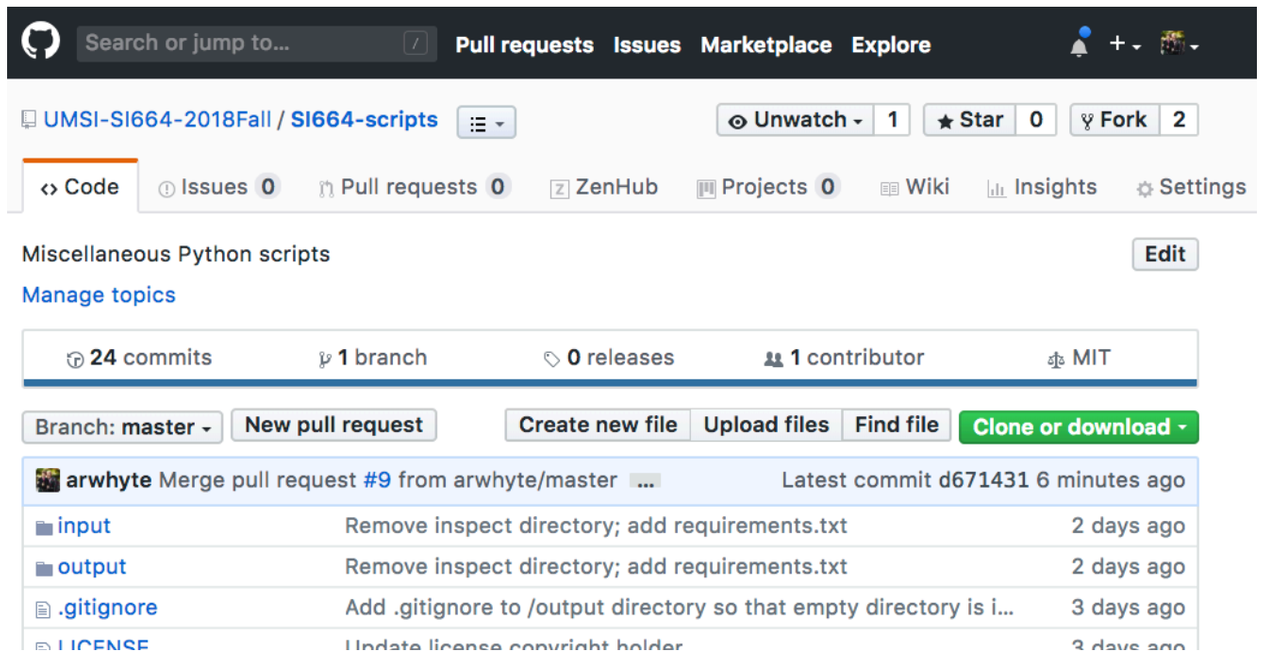
New IA

Sangeetha Mandayam Gomatam

New IA

Canvas: Discussion Tool

Github: SI664 scripts



https://github.com/UMSI-SI664-2018Fall/SI664-scripts

README.md

SI664-scripts

Miscellaneous Python scripts

Install

Either fork this repo and then clone to your working directory or download a *.zip file of the code. Create a virtual environment and then run `pip` to install package dependencies listed in the `requirements.txt` file.

macOS

```
$ cd path/to/SI664/scripts
$ source venv/bin/activate
(venv) $ pip install -r requirements.txt
```

Windows

```
> cd path/to/SI664/scripts
> venv\Scripts\activate
(venv) > pip install -r requirements.txt
```

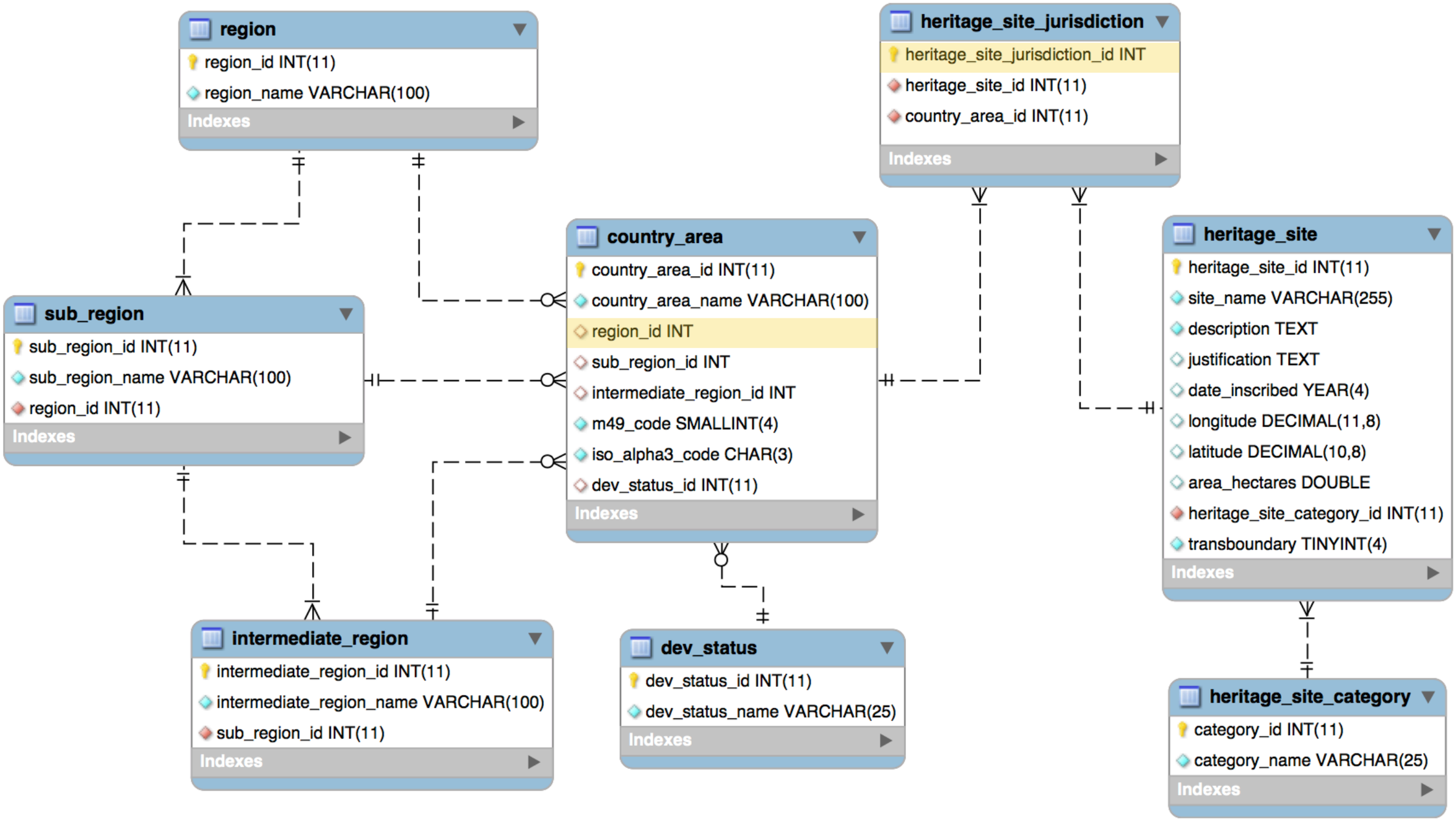

UMSI

data model

(UNSD/UNESCO heritage sites)

UMSI

UNSD/UNESCO heritage sites (Django)



admin site

Model: Location __str__(self)

```
class Location(models.Model):
    """
    New model based on Mtg 5 refactoring of the database.
    """
    location_id = models.AutoField(primary_key=True)
    # other fields

    class Meta:
        managed = False
        db_table = 'location'
        ordering = ['field 1', 'field 2', etc.]
        verbose_name = 'UNSD M49 Location Hierarchy'
        verbose_name_plural = 'UNSD M49 Location Hierarchies'

    def __str__(self):
        return '{} {} {} {}'.format(
            self.w,
            self.x if self.x else '',
            self.y if self.y else '',
            self.z if self.z else '')
```

Django needs
a string returned

functions & operators

SQL: Chinese heritage sites

```
SELECT reg.region_name,sub.sub_region_name,
        ca.country_area_name, hs.site_name, hsc.category_name
 FROM heritage_site hs
        LEFT JOIN heritage_site_jurisdiction hsj
            ON hs.heritage_site_id = hsj.heritage_site_id
        LEFT JOIN country_area ca
            ON hsj.country_area_id = ca.country_area_id
        LEFT JOIN heritage_site_category hsc
            ON hs.heritage_site_category_id = hsc.category_id
        LEFT JOIN region reg
            ON ca.region_id = reg.region_id
        LEFT JOIN sub_region sub
            ON ca.sub_region_id = sub.sub_region_id
WHERE ca.country_area_name LIKE 'China%'
ORDER BY hs.site_name;
```

ORM: Chinese heritage sites

```
>>> hs = HeritageSite.objects
.select_related('heritage_site_category')
.filter(country_area__country_area_name__startswith = ‘China')
.values_list('country_area__region__region_name',
‘country_area__sub_region__sub_region_name',
'country_area__country_area_name',
'site_name',
'heritage_site_category__category_name')

>>> for s in hs:
…        print(s[0],s[1],s[2],s[3],s[4])
```

SQL: COUNT(*)

```
SELECT COUNT(*) AS `site count` FROM heritage_site;

+------------+
| site count |
+------------+
|       1092 |
+------------+
1 row in set (0.00 sec)
```

SQL: GROUP_BY, COUNT(*)

Return count of heritage sites by region

```
SELECT r.region_name AS `region`, COUNT(*) AS `heritage sites`
  FROM heritage_site hs
        LEFT JOIN heritage_site_jurisdiction hsj
              ON hs.heritage_site_id = hsj.heritage_site_id
        LEFT JOIN country_area ca
              ON hsj.country_area_id = ca.country_area_id
        LEFT JOIN location l
              ON ca.location_id = l.location_id
        LEFT JOIN region r
              ON l.region_id = r.region_id
  GROUP BY r.region_name
  ORDER BY `heritage sites` DESC;
```

```
+----------+----------------+
| region   | heritage sites |
+————————+-----------------+
| Europe   |            488 |
| Asia     |            314 |
| Americas |            191 |
| Africa   |            144 |
| Oceania  |             30 |
| NULL     |              1 |
+----------+----------------+
6 rows in set (0.02 sec)
```

ORM: .count(), Count

```
>>> from heritagesites.models import HeritageSite
>>> hs = HeritageSite.objects.all().count()
>>> print(hs)
1092
```

```
>>> from heritagesites.models import HeritageSite
>>> from django.db.models import Count
>>> hs = HeritageSite.objects.annotate(site_count=Count('site_name'))
>>> hs.count()
1092
```


UMSI

ORM: aggregate

SQL equivalent: SUM(), AVG(), MIN(), MAX() without a GROUP BY

```
>>> from heritagesites.models import HeritageSite
>>> from django.db.models import Max
>>> hs = HeritageSite.objects.all().aggregate(max_hectares=Max('area_hectares'))
>>> print(hs)
{'max_hectares': 40825000.0}
```

```
SELECT MAX(area_hectares) FROM heritage_site;
+-------------------+
| MAX(area_hectares) |
+-------------------+
|          40825000 |
+-------------------+
1 row in set (0.00 sec)
```


UMSI

ORM: annotation

SQL equivalent: GROUP BY on the column id

```
>>> from heritagesites.models import HeritageSite
>>> from django.db.models import Count
>>> hs = HeritageSite.objects.all()
.values('heritage_site_category_id')
.annotate(count=Count('heritage_site_category_id'))
>>> hs.count()
3
```

```
mysql> SELECT heritage_site_category_id, COUNT(heritage_site_category_id)
    -> FROM heritage_site
    -> GROUP BY heritage_site_category_id;
+--------------------------+--------------------------------+
| heritage_site_category_id | COUNT(heritage_site_category_id) |
+--------------------------+--------------------------------+
|                        1 |                            845 |
|                        2 |                            209 |
|                        3 |                             38 |
+--------------------------+--------------------------------+
3 rows in set (0.00 sec)
```

ORM: annotate (GROUP BY), F object

Return count of developed vs developing countries/areas in Sub-Saharan Africa

```
>>> from heritagesites.models import Location, Region, CountryArea, DevStatus
>>> from django.db.models import Count
>>> from django.db.models import F

>>> loc = Location.objects
.values(sub_region_name = F('sub_region__sub_region_name'), dev_status =
F('countryarea__dev_status__dev_status_name'))
.annotate(count=Count('countryarea__dev_status__dev_status_name'))
.filter(sub_region__sub_region_name = 'Sub-Saharan Africa')
.order_by('countryarea__dev_status__dev_status_name')

>>> for l in loc:
...     print(l)
...
...
{'sub_region_name': 'Sub-Saharan Africa', 'dev_status': 'Developing', 'count': 53}
```


UMSI

SQL: MAX() with a subquery

Return the largest heritage site by area (hectares)

```
SELECT r.region_name AS `region`, sr.sub_region_name AS `subregion`,
        ca.country_area_name AS `country / area`,
        hs.site_name AS `heritage site`, hs.area_hectares AS `area (hectares)`
 FROM heritage_site hs
        LEFT JOIN heritage_site_jurisdiction hsj
                ON hs.heritage_site_id = hsj.heritage_site_id
        LEFT JOIN country_area ca
                ON hsj.country_area_id = ca.country_area_id
        LEFT JOIN location l
                ON ca.location_id = l.location_id
        LEFT JOIN region r
                ON l.region_id = r.region_id
        LEFT JOIN sub_region sr
                ON l.sub_region_id = sr.sub_region_id
WHERE hs.area_hectares = (SELECT MAX(hs1.area_hectares)
                          FROM heritage_site hs1)\G
```

SQL: MAX() with a subquery

Return the largest heritage site by area (hectares)

```
SELECT r.region_name AS `region`, sr.sub_region_name AS `subregion`,
        ca.country_area_name AS `country / area`,
        hs.site_name AS `heritage site`, hs.area_hectares AS `area (hectares)`
 FROM heritage_site hs
        LEFT JOIN heritage_site_jurisdiction hsj
                ON hs.heritage_site_id = hsj.heritage_site_id
        LEFT JOIN country_area ca
                ON hsj.country_area_id = ca.country_area_id
        LEFT JOIN location l
                ON ca.location_id = l.location_id
        LEFT JOIN region r
                ON l.region_id = r.region_id
        LEFT JOIN sub_region sr
                ON l.sub_region_id = sr.sub_region_id
WHERE hs.area_hectares = (SELECT MAX(hs1.area_hectares)
                          FROM heritage_site hs1)\G
```


UMSI

SQL: MAX() with a subquery

Return the largest heritage site by area (hectares)

```
*************************** 1. row ***************************
region: Oceania
subregion: Micronesia
country / area: Kiribati
heritage site: Phoenix Islands Protected Area
area (hectares): 40825000
1 row in set (0.01 sec)
```

SQL: GROUP BY, GROUP_CONCAT

Return counts of country/area regional affiliations and list the countries/areas

```
SELECT l.region_id, r.region_name, l.sub_region_id, sr.sub_region_name,
        l.intermediate_region_id, ir.intermediate_region_name, COUNT(*) AS count,
       GROUP_CONCAT(ca.country_area_name SEPARATOR ', ') AS `countries / areas`
  FROM country_area ca
        LEFT JOIN location l
              ON ca.location_id = l.location_id
        LEFT JOIN region r
              ON l.region_id = r.region_id
        LEFT JOIN sub_region sr
              ON l.sub_region_id = sr.sub_region_id
        LEFT JOIN intermediate_region ir
              ON l.intermediate_region_id = ir.intermediate_region_id
GROUP BY l.region_id, l.sub_region_id, l.intermediate_region_id
ORDER BY l.region_id, l.sub_region_id, l.intermediate_region_id\G
```

SQL: GROUP BY, GROUP_CONCAT

Return counts of country/area regional affiliations and list the countries/areas

```
SELECT l.region_id, r.region_name, l.sub_region_id, sr.sub_region_name,
        l.intermediate_region_id, ir.intermediate_region_name, COUNT(*) AS count,
       GROUP_CONCAT(ca.country_area_name SEPARATOR ', ') AS `countries / areas`
 FROM country_area ca
        LEFT JOIN location l
                ON ca.location_id = l.location_id
        LEFT JOIN region r
                ON l.region_id = r.region_id
        LEFT JOIN sub_region sr
                ON l.sub_region_id = sr.sub_region_id
        LEFT JOIN intermediate_region ir
                ON l.intermediate_region_id = ir.intermediate_region_id
GROUP BY l.region_id, l.sub_region_id, l.intermediate_region_id
ORDER BY l.region_id, l.sub_region_id, l.intermediate_region_id\G
```

back ticks for quoting

print out in vertical output format

UMSI

SQL: vertical output format (\G)

Return counts of country/area regional affiliations and list the countries/areas

```
*************************** 2. row ***************************
region_id: 1
region_name: Africa
sub_region_id: 8
sub_region_name: Northern Africa
intermediate_region_id: NULL
intermediate_region_name: NULL
count: 7
countries/areas: Tunisia, Algeria, Sudan, Egypt, Morocco, Western Sahara, Libya
```

SQL: CONCAT, IFNULL

Compare country_area/location table regional affiliations vs ye olde country_area internal regional foreign keys approach

```
SELECT CONCAT('CA: ',
              ca.country_area_name, ' ',
              IFNULL(ca.region_id, 0), ' ',
              IFNULL(ca.sub_region_id, 0), ' ',
              IFNULL(ca.intermediate_region_id, 0)),
           CONCAT('LOC: ',
              IFNULL(l.region_id, 0), ' ',
              IFNULL(l.sub_region_id, 0), ' ',
              IFNULL(l.intermediate_region_id, 0))
  FROM country_area ca
           LEFT JOIN location l
                  ON ca.location_id = l.location_id
WHERE IFNULL(ca.region_id, 0) = IFNULL(l.region_id, 0)
   AND IFNULL(ca.sub_region_id, 0) = IFNULL(l.sub_region_id, 0)
   AND IFNULL(ca.intermediate_region_id, 0) = IFNULL(l.intermediate_region_id, 0)
ORDER BY ca.region_id, ca.sub_region_id, ca.intermediate_region_id, ca.country_area_name\G
```

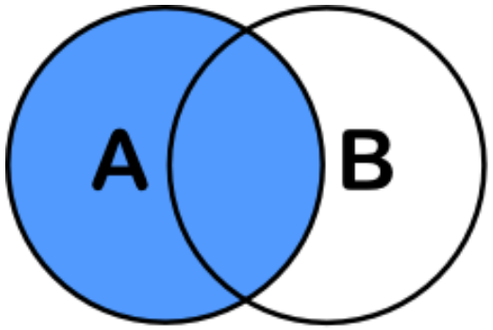
SQL: vertical output format (\G)

Compare country_area/location table regional affiliations vs ye olde country_area internal regional foreign keys approach

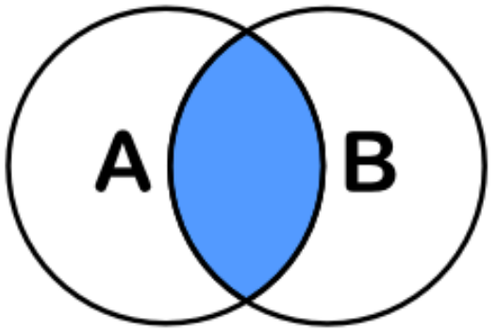
```
*************************** 249. row ***************************
CONCAT('CA: ',
  ca.country_area_name, ' ',
  IFNULL(ca.region_id, 0), ' ',
  IFNULL(ca.sub_region_id, 0), ' ',
  IFNULL(ca.intermediate_region_id, 0)): CA:Wallis and Futuna Islands 5 11 0
CONCAT('LOC: ',
  IFNULL(l.region_id, 0), ' ',
  IFNULL(l.sub_region_id, 0), ' ',
  IFNULL(l.intermediate_region_id, 0)): LOC: 5 11 0
```

match

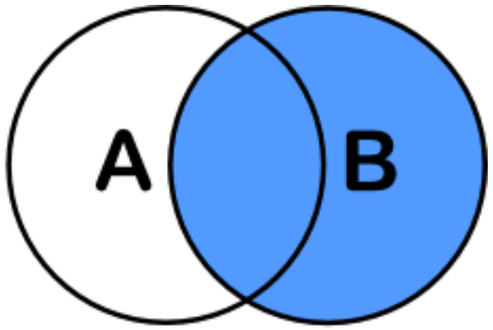
SQL JOINS



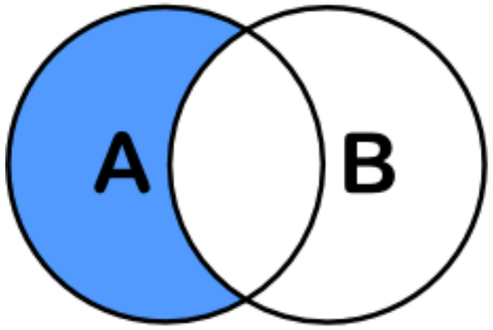
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key



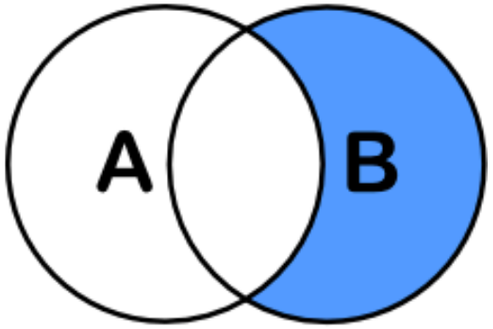
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key



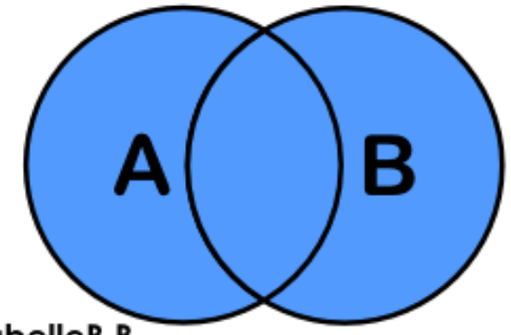
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key



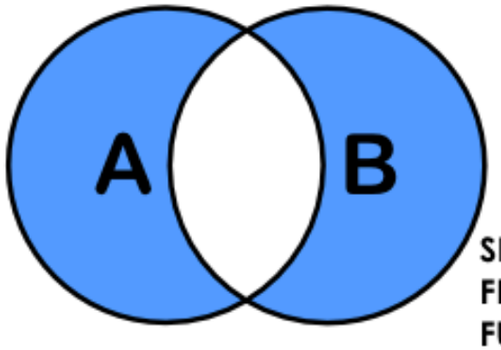
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL



SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL



SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key



SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL

UMSI

finis

UMSI