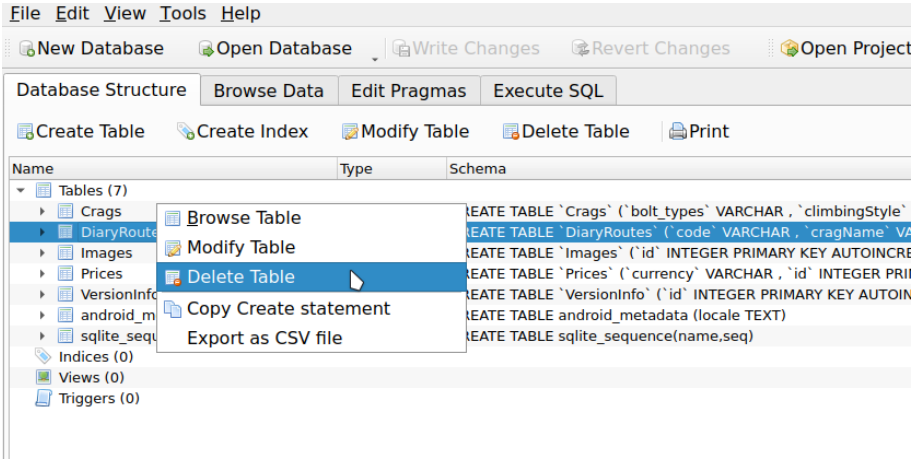
Razvoj in upravljanje programskih sistemov

Sprint 2 - Poročilo

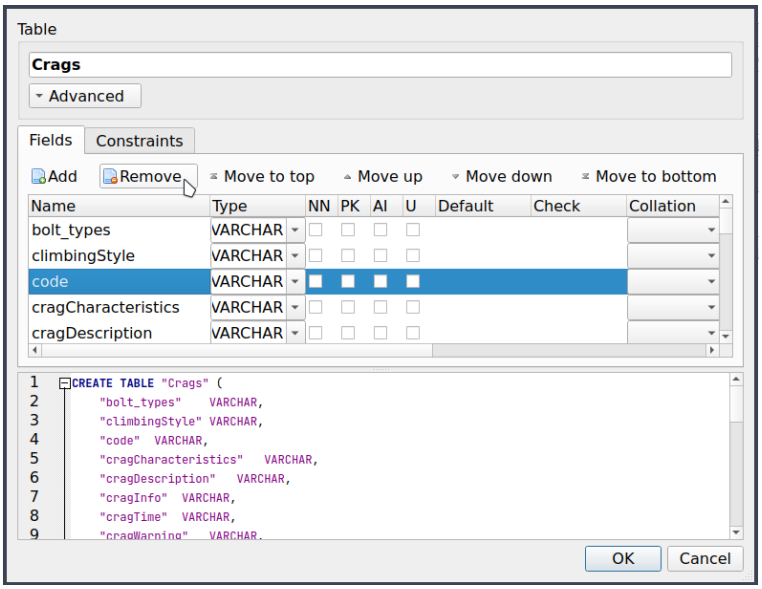
Urban Prah

7. november 2022

S pomočjo programa `sqlitebrowser` smo preuredili obstoječo bazo tako, da smo najprej odstranili odvečne podatke. Mednje so spadale table s slikami, cenami, uporabnikov dnevnik.



Odstranili smo tudi nekaj stolpcev kot so različni identifikatorji, prevodi, itd.

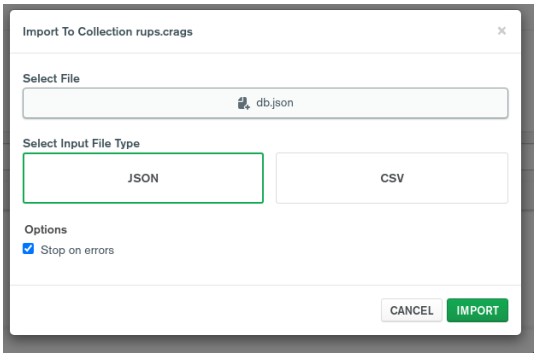


Podatke table smo izvozili v datoteko CSV in jih pretvorili v format JSON.

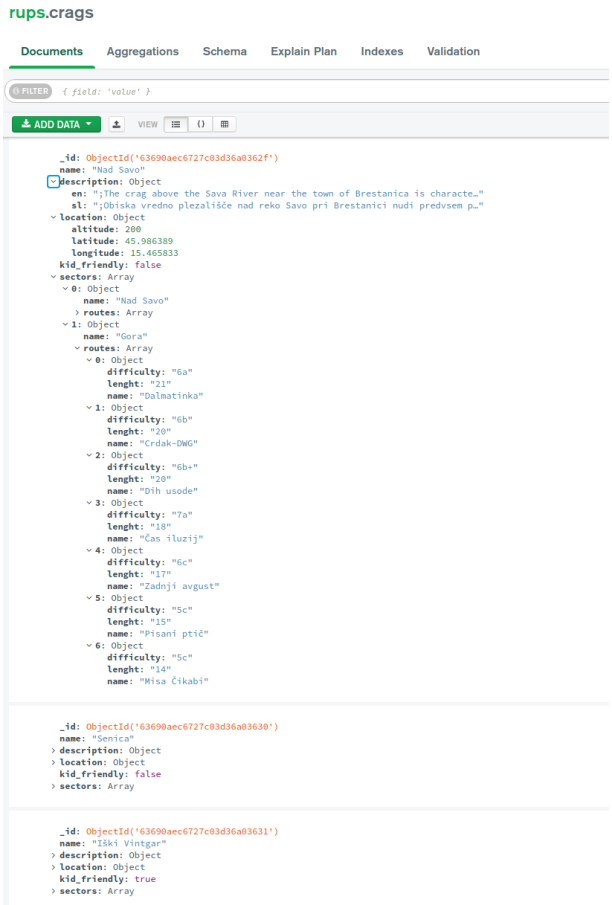
```
rows = []
with open("./Crags.csv") as csvfile:
    sr = csv.reader(csvfile, delimiter=",", quotechar="'")
    for s in sr:
        rows.append(s)
for i in range(1, len(rows)):
    for k in range(len(rows[i])):
        if "{" in rows[i][k]:
            rows[i][k] = json.loads(rows[i][k])
crags = []
for i in range(1, len(rows)):
    crag = {}
    crag["name"] = rows[i][7]
    d = rows[i][0]
    del d["de"]
    del d["it"]
    d = { key: d[key] + ";" + rows[i][1][key] + ";" + rows[i][2][key] + ";"
          + rows[i][3][key] + ";" + rows[i][4][key] for key in d.keys() }
    crag["description"] = d
    crag["location"] = rows[i][6]
    crag["kid_friendly"] = rows[i][5] == "1"
    crag["sectors"] = rows[i][9]
    crags.append(crag)
for c in range(len(crags)):
    for s in range(len(crags[c]["sectors"])):
        try: del crags[c]["sectors"][s]["code"]
        except: pass
        try: del crags[c]["sectors"][s]["crag_code"]
        except: pass
        try: del crags[c]["sectors"][s]["images"]
        except: pass
        try: del crags[c]["sectors"][s]["order"]
        except: pass
        try: del crags[c]["sectors"][s]["route_count"]
        except: pass
for c in crags:
    for s in c["sectors"]:
        for r in s["routes"]:
            if "parentRouteCode" in r.keys():
                del r
for c in range(len(crags)):
    for s in range(len(crags[c]["sectors"])):
        for r in range(len(crags[c]["sectors"][s]["routes"])):
            try: del crags[c]["sectors"][s]["routes"][r]["code"]
            except: pass
            try: del crags[c]["sectors"][s]["routes"][r]["mapping"]
            except: pass
            try: del crags[c]["sectors"][s]["routes"][r]["order"]
            except: pass
            try: del crags[c]["sectors"][s]["routes"][r]["tags"]
            except: pass
            d = crags[c]["sectors"][s]["routes"][r]["difficulty"]
            crags[c]["sectors"][s]["routes"][r]["difficulty"] = "{}{}{}".format(
                d["number"], d["letter"], d["mark"]
            )
with open("db.json", "w") as f:
    f.write(json.dumps(crags, indent=2))
```

Skripta preoblikuje obstoječe podatke v nam zaželeno obliko ter jih izvozi v datoteko db.json. Sledi namestitev podatkovne baze MongoDB in uvoz podatkov.

Za uvoz smo uporabili uporabniški vmesnik `mongodb-compass`.



Slika prikazuje vnose v podatkovni bazi.



Načrtovali smo uporabo podatkovne baze MySQL, vendar smo tekom razvoja ugotovili da relacijska baza ni ustrezna za hrambo našega tipa podatkov. Spremeba ne vpliva na nadalnji razvoj.

Prihodnji teden sledi razširitev vnosov v podatkovni bazi in njihovih lastnosti. Odkriti bomo poskušali čim več obstoječih virov podobnih obstoječemu.