**SPOSOBNOSTI IN OMEJITVE NOCODB-ja**

Po dokumentaciji na <https://docs.nocodb.com/> nocodb podpira naslednje storitve:

Kar se tiče samih vnosov podatkov v tabelo NocoDB omogoča:

-kreiranje svoje tabele v NocoDB,

- import prek CSV-ja, Airtable, JSON in Excel-a, vendar le 3 datoteke po 25Mb naenkrat

- povezava z zunanjo bazo, kot je MySQL, SQLite in PostgreSQL, midva sva se odločila za to opcijo, saj je bila edina izvedljiva.

Za analizo podatkov v tabeli podpira naslednje storitve:

1. **Skrivanje ali prikaz atributa v tabeli.**
2. **Filtriranje**

Omogoča filtriranje po več pogojih, vključuje tudi and in or.

Za tekstovna polja ima naslednje opcije:

**is equal** – Filtrira zapise, kjer je vrednost enaka določeni vrednosti.

**is not equal** – Filtrira zapise, kjer vrednost ni enaka določeni vrednosti.

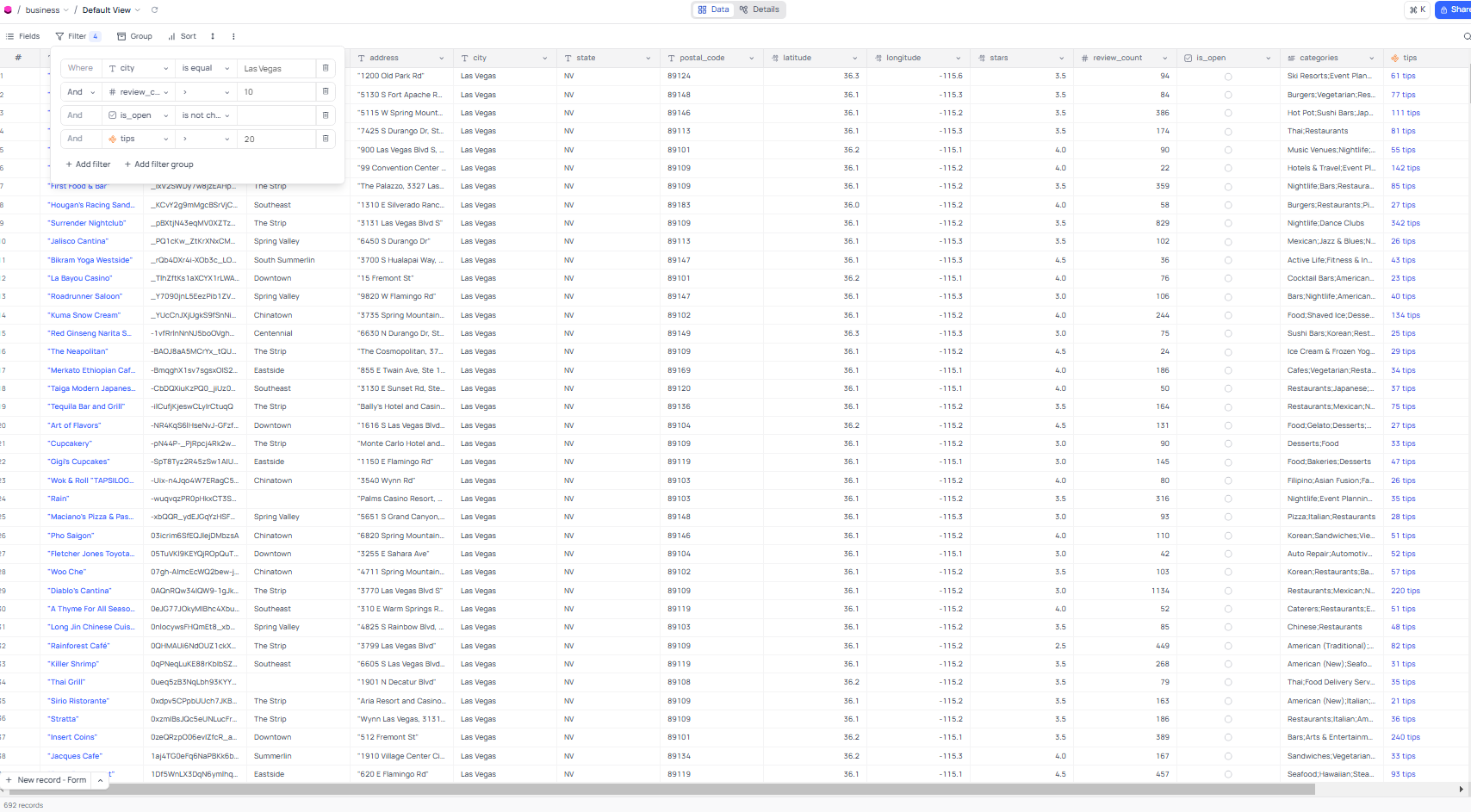
**is like** – Filtrira zapise, kjer vrednost vsebuje določeno podniz (podpira vzorčenje).

**is not like** – Filtrira zapise, kjer vrednost ne vsebuje določenega podniza.

**is blank** – Filtrira zapise, kjer je vrednost prazna.

**is not blank** – Filtrira zapise, kjer vrednost ni prazna.

Za številke omogoča polja: =, !=, >, >=, <, <=, is blank.



Primer filtriranja.

1. **Sortiranje**

Podobno kot pri filtriranju, omogoča NocoDB **sortiranje** podatkov po:

* **Abecednem vrstnem redu**: Za besedilna polja (od A do Ž ali obratno).
* **Številčnem vrstnem redu**: Za številčna polja (od najmanjše do največje vrednosti ali obratno).

1. **Grupiranje**

NocoDB omogoča funkcijo **grupiranja podatkov** znotraj tabel:

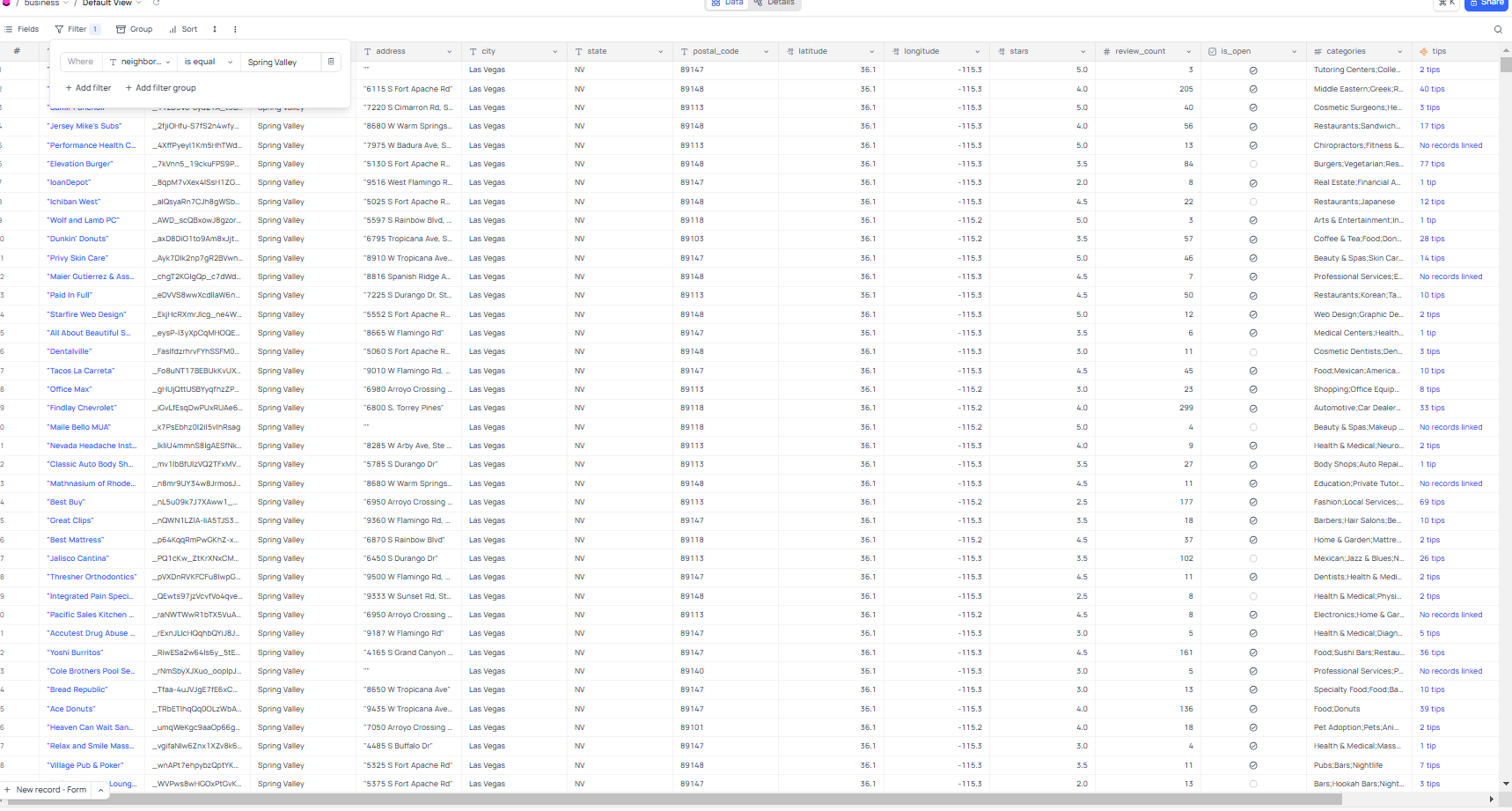
**Grupiranje po enemu ali več stolpcih**:

Združi zapise po enemu ali več atributih.

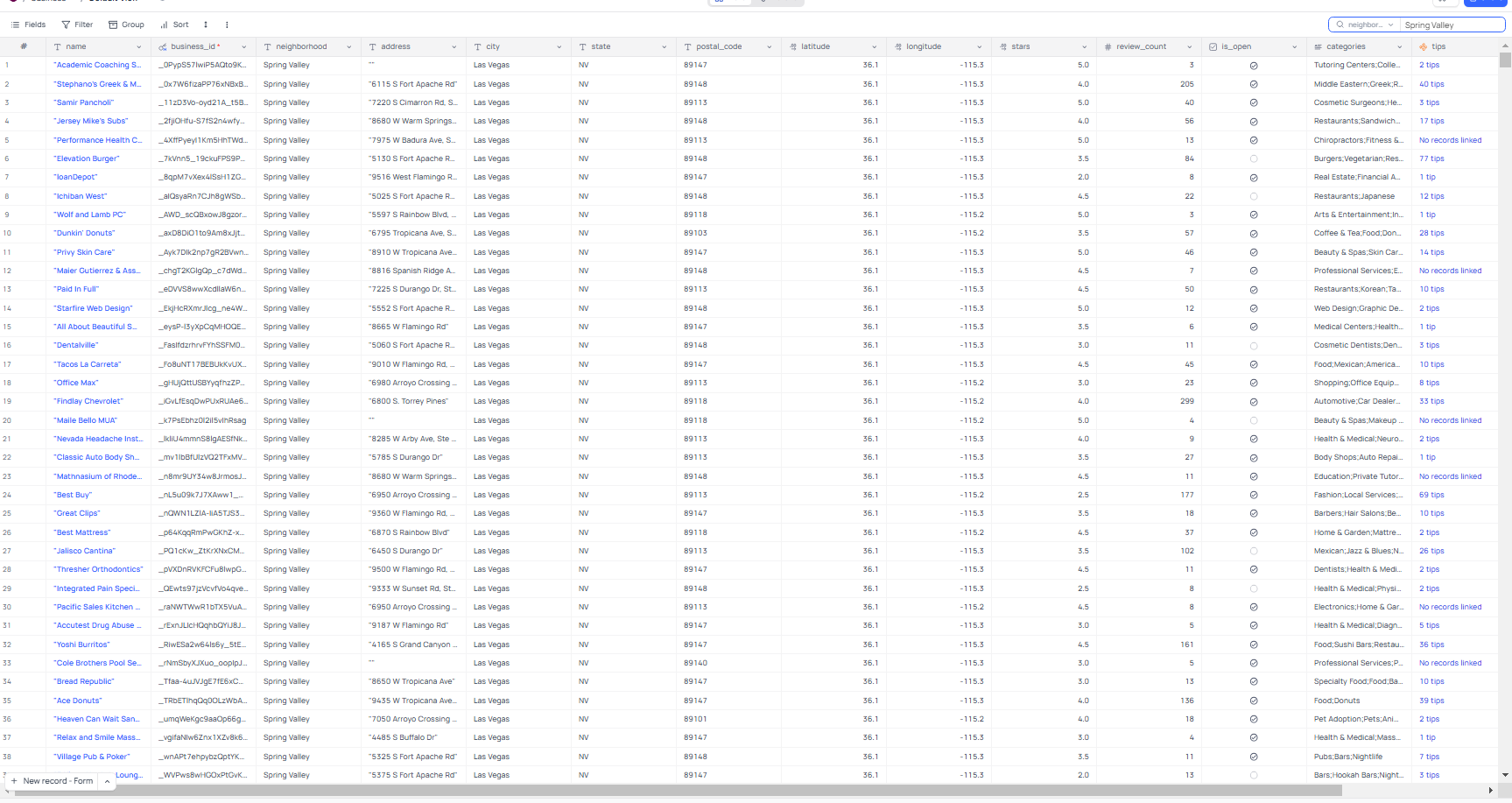
* **Preglednost podatkov**:
  + Združene skupine so lahko razširjene ali skrčene, kar omogoča lažji pregled.
* **Štetje zapisov**:
  + Vsaka skupina prikazuje število zapisov, ki jih vsebuje.

Omogoča tudi search funkcijo, ki deluje na isti način kot filtriranje:

Primer filtriranja



Primer iskanja:



Details zavihek ima podatke o tabeli:

**4.1 Fields**

* Prikazuje vse stolpce (atribute) v tabeli.
* Daje informacije o tipu podatkov v posameznem stolpcu, kot so:
  + Besedilo (String)
  + Številke (Number)
  + Datum (Date)
  + Boolean (True/False)
  + Relacije (Links z drugimi tabelami)
* Omogoča dodajanje, urejanje ali brisanje stolpcev.

**4.2 Relations**

* Prikazuje povezave med tabelo in drugimi tabelami.
* Vse relacije, ki so bile ustvarjene (npr. **One-to-One**, **One-to-Many**, **Many-to-Many**), so prikazane tukaj, vendar moraš miško dati na povezavo, če želiš videti kakšna je.
* Omogoča enostavno upravljanje relacij in pregled, kako so podatki povezani.

4.3. **API snippet**

* Vsebuje avtomatsko generirane primere za API klice, ki jih lahko uporabiš, da se povežeš s tabelo.
* Podani primeri vključujejo:
  + **GET**: Pridobivanje podatkov iz tabele.
  + **POST**: Dodajanje novih zapisov v tabelo.
  + **PUT**: Posodabljanje obstoječih zapisov.
  + **DELETE**: Brisanje zapisov.

Orodja, za katera so vnaprej pripravljeni primeri so: Javascript, Python, Ruby, PHP, Java, C, Shell, Node in NocoDB-sdk.

* + 1. **Webhooks**

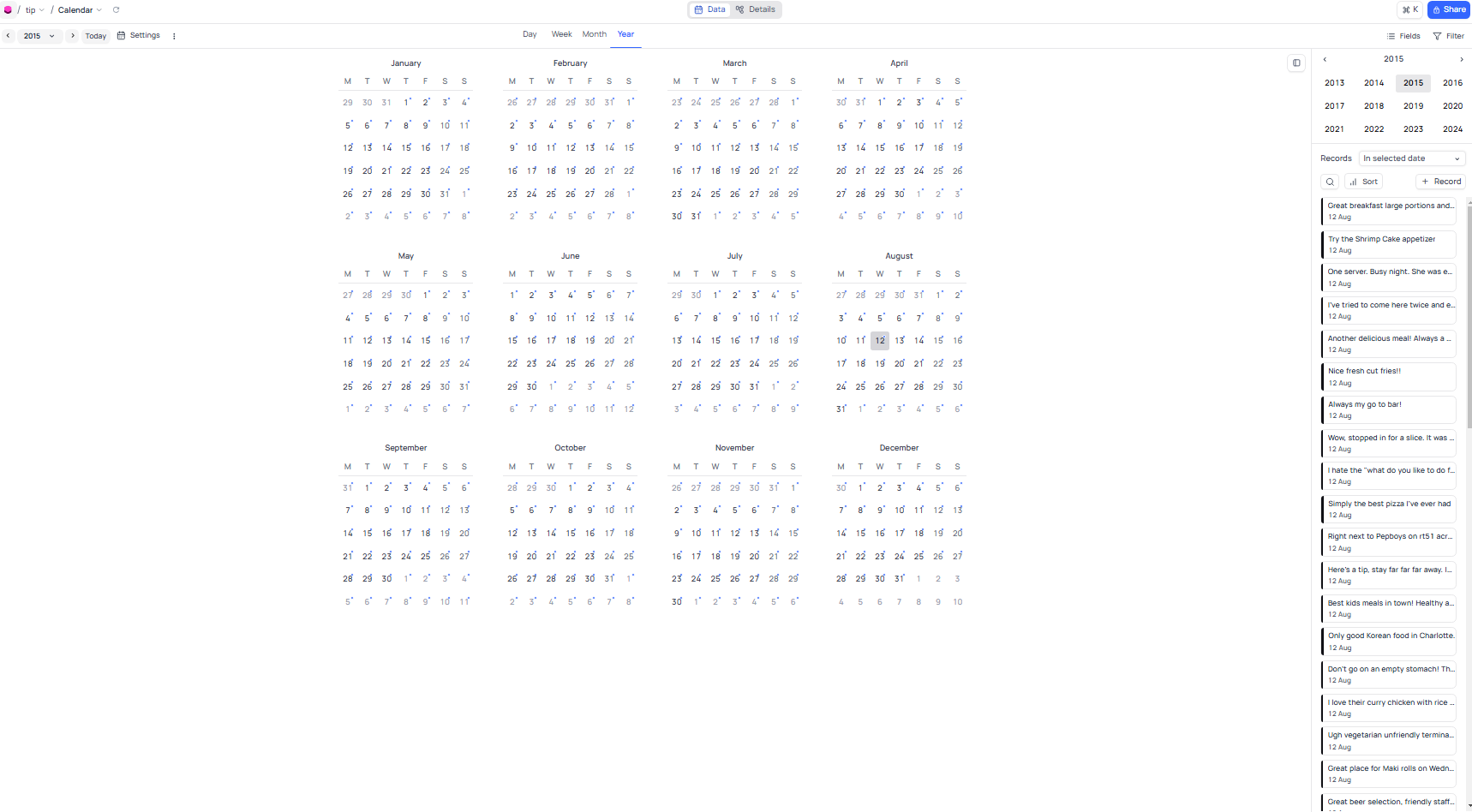
Morda še najbolj zanimiva stvar do zdaj, **webhooki** so v bistvu bazni prožilci, ki jih lahko vgradiš kar v NocoDB in jih uporabiš za avtomatizacijo procesov ter integracijo z zunanjimi sistemi. Omogočajo:

* **Avtomatizacija**: Webhooki se samodejno sprožijo ob določenih dogodkih, kot so:
  + Dodajanje novega zapisa (Insert).
  + Posodabljanje obstoječega zapisa (Update).
  + Brisanje zapisa (Delete).
  + Vnos ali posodobitev več zapisov naenkrat (Bulk Insert/Update/Delete).
  + Ročna sprožitev (Manual Trigger) preko gumbov v tabeli.
* **Prilagodljivost**:
  + Možnost definiranja pogojev, pod katerimi se sproži webhook (npr. le, če je status "Complete").
  + Podpora za različne metode HTTP zahtevkov: **GET**, **POST**, **PUT**, **DELETE**, **PATCH**.
  + Nastavitev **Headerjev** in **parametrov** za prilagoditev zahteve.

**Kako delujejo pogoji za sprožitev?**

* Webhook se sproži samo, če konfigurirani pogoj prej **ni bil izpolnjen**, po posodobitvi zapisa pa **je izpolnjen**.
  + Primer: Webhook se sproži samo, če se status spremeni iz "Pending" v "Complete".
* Pogoji lahko vključujejo **AND** in **OR** operatorje za kompleksne logične zahteve.
  + 1. **Views**

Tako kot MySQL, NocoDB omogoča poglede. Ima več stilov za poglede: Grid, Form, Gallery, Kanban, Date



Tukaj je primer za Date: pokaže vse reviewe napisane 12.8.2015.

* + 1. **Dodatne opcije pri poljih**

V NocoDB so na voljo različna polja, ki omogočajo različne načine upravljanja s podatki. Tukaj je nekaj primerov teh dodatnih polj:

**Lookup**: Deluje podobno kot JOIN v MySQL, ko uporabimo omogočimo NocoDB povezavo z izbrano tabelo, edino kar je malce nadležno, je da lahko le eno povezavo na field (izraz za stolpec v NocoDB) narediš, ampak ker ponavadi iščeš le 2-3 podatke iz druge tabele, je ta funkcija precej uporabna, v spodnjem screenshotu sem v tabelo reviews dodal review\_count in useful.



**Formula:** Omogoča izvajanje izračunov na podlagi drugih polj v tabeli, nekaj bolj zanimiv možnosti za stringe so: **SUBSTR,** , SEARCH. JSON\_EXTRACT, REGEX\_REPLACE in URL. Seveda podpira tudi numerične funkcije, nekatere bolj zanimive so: POWER, SQRT, MIN in MAX, ima pa tudi funkcije za Date. Večina funkcij je iz SQL, nobena ni dodana, le katero ime je spremenjeno, da je bolj intuitivno (SEARCH namesto INSTR). Podpira tudi IF stavke in ima tudi SWITCH-CASE.

**Rollup** omogoča izvajanje agregacijskih funkcij nad povezanimi tabelami. To pomeni, da lahko izračunate vrednosti, kot so seštevek, povprečje, število zapisov, najmanjša ali največja vrednost, na podlagi podatkov v povezanih tabelah. Primer rollup-a bom pokazal kasneje, saj je ena bolj zanimivih lastnosti NocoDB-ja. Uporaba spominja na GROUP BY stavek z dodatno funkcijo, kot AVG ali COUNT.

Druge možnosti: Poleg prejšnjih možnosti NocoDB tudi omogoča:

-Qr code: Pusti ti generirati Qr code iz string-a. Uporabno za stvari kot so telefonska številka, gmail ali URL.

- Barcode: Podobno kot QR code.

- Buttons: Omogoča interaktivne gumbe znotraj tabel, ki lahko odprejo kakšen link ali pa izvedejo webhook.

**API**

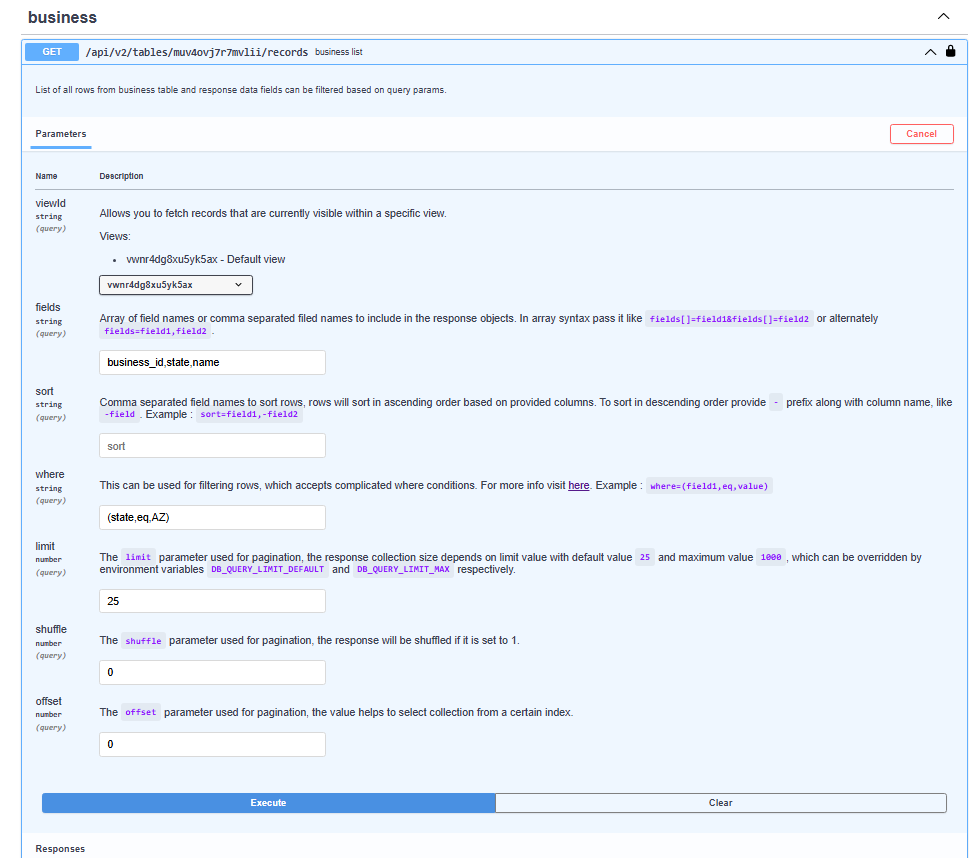
NocoDB omogoča enostavno uporabo API-jev za interakcijo z podatki, kar omogoča, izvajanje različnih operacij na tabelah in povezavah. Vse operacije so dostopne preko REST API-ja. V bistvu delaš psevdo-sql stavke, edina razlika je, da ti nocoDB razloži kako uporabiti vsako polje, kot bom prilepil v spodnjem screenshotu. Zahtevke, ki jih API omogoča so naslednji:

1. **GET**:
   * Uporablja se za pridobivanje podatkov.
   * Na primer, /api/v2/tables/{table\_name}/records omogoča pridobitev vseh zapisov iz tabele.
   * Lahko vključite specifične filtre, da pridobite le določene zapise.
2. **POST**:
   * Uporablja se za ustvarjanje novih zapisov.
   * Na primer, /api/v2/tables/{table\_name}/records vam omogoča, da dodate nov zapis v tabelo.
3. **PATCH**:
   * Uporablja se za posodabljanje obstoječih zapisov.
   * Na primer, /api/v2/tables/{table\_name}/records omogoča posodobitev specifičnega zapisa v tabeli.
4. **DELETE**:
   * Uporablja se za brisanje zapisov.
   * Na primer, /api/v2/tables/{table\_name}/records/{recordId} omogoča brisanje določenega zapisa z določenim ID-jem.
5. **Link and Unlink Records**:
   * Določa povezovanje in odstranjevanje povezav med zapisi v različnih tabelah.
   * Na primer, /api/v2/tables/{table\_name}/links/{linkFieldId}/records omogoča povezovanje zapisov med tabelami
   * Pomoč pri štetju števila zapisov v tabeli z uporabo /api/v2/tables/{table\_name}/records/count.

Vse zgoraj omenjene operacije omogočajo učinkovito manipulacijo z vašimi podatki in omogočajo integracijo z zunanjimi sistemi.

**Varnost:**

Vidimo, da so nekateri API-ji zaščiteni z ikono ključavnice, kar pomeni, da so zaščiteni s preverjanjem pristnosti (avtentifikacija) in da lahko dostopate do teh API-jev samo, če ste avtentificirani kot uporabnik z svojim tokenom, ki ga lahko vidi le uporabnik.



Poizvedba vrne 5 primerov business\_id,state,name,city, ki so v Las Vegasu.

JSON response

**{**

**"list": [**

**{**

**"business\_id": "\_\_3qOwWFBUE8mdOToI7YrQ",**

**"name": "\"Custom Kings\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_\_IsqCZAF9YTcvKPKj2dZg",**

**"name": "\"Z Gallerie\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_\_NdJ7B-wxGyz\_V-X0NTOA",**

**"name": "\"Load 4 Less\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_\_pP-ARewXjwdp5HF6SHVQ",**

**"name": "\"Golden Palace Massage\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_\_vwoTU6B1r0FvT1CeAHVw",**

**"name": "\"Welcome Dvd\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_-3aZXuecjOSjramI1IBkA",**

**"name": "\"Fountain Grille Restaurant\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_-481O5q5--sJeHC3jQhdw",**

**"name": "\"Nevada Suzuki\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_-DzOH5j-PbRrMf\_Nm6w-Q",**

**"name": "\"Royce'\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_-gU\_Go0poUt96LwgGoXTw",**

**"name": "\"Iron Trainer Personal Training\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**},**

**{**

**"business\_id": "\_-QrdVYJbebaC2qschZ6iQ",**

**"name": "\"Little Caesars Pizza\"",**

**"city": "Las Vegas",**

**"state": "NV"**

**}**

**],**

**"pageInfo": {**

**"totalRows": 26775,**

**"page": 1,**

**"pageSize": 10,**

**"isFirstPage": true,**

**"isLastPage": false**

**}**

**}**