Engineer

PostGIS

utvidelsen som gir PostgreSQL geometri

Hva det er

PostGIS er en utvidelse med åpen kildekode til databaseserveren PostgreSQL. Den lar en kolonne holde en form, f.eks. en leveringsadresse, en elv eller en eiendomsgrense, og lar vanlig SQL spørre hvor formene ligger i forhold til hverandre. Prosjektet startet som et forskningsprosjekt hos Refractions Research, et konsulentselskap innen GIS og databaser, og utvikles og utgis nå som et selvstendig prosjekt med åpen kildekode.

Utvidelsen installeres per database med én setning, `CREATE EXTENSION postgis;`, og tre ting følger med.

- **Romlige typer.** `geometry` beskriver former på et flatt plan, i det koordinatsystemet dataene ble samlet inn i. `geography` beskriver dem på jordens krumme overflate og måler i meter.
- **Romlige funksjoner.** En stor familie funksjoner med navn som begynner på `ST_`: `ST_Intersects` og `ST_Contains` for forhold mellom former, `ST_Area` og `ST_Length` for måling, `ST_Transform` for å flytte data mellom koordinatsystemer og `ST_AsGeoJSON` for å gi resultatet videre til et webkart.
- **Romlige indekser.** PostGIS bygger et R-tre oppå GiST-rammeverket i PostgreSQL, og støtter også SP-GiST og blokkintervallindekser (BRIN) for bestemte typer data.

Formene følger Simple Features-modellen som Open Geospatial Consortium har publisert, og som er tatt inn som ISO 19125: punkter, linjer og polygoner, flerdelte varianter av dem og samlinger av blandede typer. Hver lagrede verdi har en SRID, et tall som peker på en rad i tabellen `spatial_ref_sys` og sier hva koordinatene betyr. SRID 4326 betyr lengdegrad og breddegrad i grader på datumet WGS 84, systemet GPS rapporterer i. Et projisert system måler derimot i meter øst og nord for et fast origo, og er bare nøyaktig innenfor området det ble laget for.

Rasterdata, topologi og noen 3D-operasjoner kommer som egne utvidelser, `postgis_raster`, `postgis_topology` og `postgis_sfcgal`, slik at en database bare bærer de delene den faktisk bruker.

Bruk begrepet i en beslutning

Bruk «Hva er PostGIS?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Alt som skjer på et sted — fartøy, kjøretøy, tomter, sensorer — lagt på et kart du kan søke i, måle på og dele.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Gjorde en geodatapipeline 50 ganger raskere — 50x raskere pipelines for geografiske data

Gjorde en geodatapipeline 50 ganger raskere Ta kontakt →

Hvorfor det er viktig

Valget PostGIS tvinger fram, er hvor stedsdata skal bo. Uten PostGIS havner koordinatene gjerne i to tallkolonner, mens selve det romlige arbeidet skjer et annet sted: i applikasjonskode som går gjennom radene én etter én for å regne ut avstander, i et GIS-program på skrivebordet som noen eksporterer til hver mandag, eller i en egen geografisk tjeneste med sin egen kopi av dataene. Hver av disse kopiene sklir ut. Med PostGIS er «hvilke kunder ligger innenfor to kilometer fra et depot som har varen inne» én spørring som kobler lagerbeholdning, kunderegister og former, i én transaksjon, med de samme sikkerhetskopiene og de samme tilgangene som alt annet i databasen.

Det betyr også at romlige data får de samme garantiene som resten av datamodellen allerede har. En CHECK-regel kan avvise et ugyldig polygon i det øyeblikket det settes inn. Sikkerhet på radnivå kan begrense et regionalt team til eiendommene i sin egen region. En import som feiler, rulles tilbake i stedet for å etterlate et halvt kartlag.

Det andre valget står mellom geometry og geography, og PostGIS-håndboken er tydelig på det. Data som holder seg innenfor én by eller ett land, lagres som regel best som `geometry` i en projeksjon valgt for området, fordi hele funksjonssettet er tilgjengelig og regningen går raskt. Data som spenner over kontinenter, eller der brukerne spør etter avstander i meter mellom punkter hvor som helst på jorden, er lettere å få riktig som `geography`. Den typen gjør regningen på den krumme overflaten selv, mot at færre funksjoner er tilgjengelige.

Noen ganger er svaret at PostGIS er mer enn jobben trenger. En app som bare lagrer lengdegrad og breddegrad per butikk og setter nåler på et kart, klarer seg med to tallkolonner. PostGIS gjør seg fortjent til plassen når spørsmålene handler om avstand, inneslutning eller overlapp, eller når dataene inneholder linjer og polygoner og ikke bare enkeltpunkter.

Slik ser det ut i praksis

PostGIS var grunnen til en migrering i ett prosjekt og arbeidshesten under en opprydding i et annet.

En kartapplikasjon som besvarte romlige spørringer i sanntid for mange brukere, kjørte på en Microsoft SQL Server-database, og latensen gikk ut over både kartet og tjenestene bak det. Løsningen var en overgang til PostgreSQL med PostGIS, med et skjema utformet for romlige data, GiST- og SP-GiST-indekser på geometri- og geografikolonnene og begrensninger på de romlige koordinatene. Mer enn 50 millioner romlige poster gikk gjennom ETL-pipelines som transformerte dem til EPSG:4326 på vei inn. Gjennomsnittlig svartid, som var 2,5 sekunder på den gamle databasen, falt til under 250 millisekunder da PostGIS besvarte spørringene, en forbedring med en faktor 10, og CPU-belastningen i backend falt med 65 %. Her er PostGIS grunnen til en migrering.

Det andre prosjektet startet med datakvalitet, ikke fart. Et stort prosjekt for geospatial analyse avdekket over 1 000 inkonsistenser i geografiske datasett og tidsseriedatasett. GDAL, QGIS og ArcGIS ble brukt til å finne og rette de romlige feilene, mens PostGIS kjørte de romlige spørringene og den romlige indekseringen, med prosedyrer i PL/pgSQL som transformerte dataene inne i PostgreSQL, og med det rensede resultatet vist i Mapbox. Kjøretiden for romlige spørringer falt med 35 %.

Hvor det går galt

Avstander kommer tilbake som bittesmå desimaltall. En spørring ber om avstanden mellom to byer og får `0.08`. Kolonnen er `geometry` i SRID 4326, så svaret er i grader, og én lengdegrad krymper fra ca. 111 kilometer ved ekvator til ingenting ved polene. Konverter til `geography` for å få svaret i meter, eller bruk `ST_Transform` til å flytte dataene over i et projisert system som passer området, før du måler.

Et radiussøk leser hver eneste rad. Tabellen har en romlig indeks, men planen viser en sekvensiell skanning. Den vanlige årsaken er et filter skrevet som `ST_Distance(a, b) < 500` . `ST_Distance` står utenfor håndbokens liste over funksjoner som bruker indeksen på egen hånd, mens `ST_DWithin` står på den. Å pakke den indekserte kolonnen inn i en konvertering eller en transformasjon har samme virkning, fordi en indeks på `geom` sier ingenting om `geom::geography` .

```
-- a geography column, indexed as stored, measured in metres
CREATE INDEX depots_geog_idx ON depots USING GIST (geog);

SELECT id FROM depots
WHERE ST_DWithin(geog, ST_MakePoint(12.57, 55.68)::geography, 500);
```

Må dataene bli liggende i en `geometry` -kolonne med SRID 4326, legger du enten til en `geography` -kolonne som denne og indekserer den, eller lager indeksen på nøyaktig det uttrykket spørringen bruker. Kjør `ANALYZE` etter en stor masseinnlasting, så planleggeren vet at indeksen er verdt å bruke.

Punktene havner i havet. Hver form beholder mønsteret sitt, men ligger på feil sted, ofte på et annet kontinent. To feil gir dette. `ST_SetSRID` setter bare en ny merkelapp på en geometri og lar tallene være i fred, så brukt der `ST_Transform` var ment, stempler den meter som grader. Eller så er lengdegrad og breddegrad byttet om et sted lenger opp i kjeden: PostGIS forventer, i likhet med GeoJSON, x før y, altså lengdegrad før breddegrad. Bruk `ST_SetSRID` bare til å erklære hva tallene allerede er, `ST_Transform` til å regne dem om, og test hver import med ett punkt du vet hvor ligger.

Areal- og overlappsberegninger kaster feil. En import fra en shapefil går gjennom, og senere stopper `ST_Intersection` med et topologiunntak. PostGIS lagrer ugyldig geometri med vilje, slik at den kan finnes og repareres, og et polygon der grensen krysser seg selv, er ugyldig. Sjekk med `ST_IsValid` , reparer med `ST_MakeValid` , som beholder hvert eneste hjørnepunkt fra inndataene, og legg til en CHECK-regel slik at en dårlig form blir avvist i døra i stedet for å bli oppdaget i en rapport.

Les mer

- PostGIS Documentation — prosjektets egen inngang til håndboken, FAQ-en og opplæringsmateriellet.

- Data Management — geometry og geography, SRID-er, gyldighetsregler og de tre indekstypene, slik håndboken definerer dem.
- Spatial Indexing — workshopkapitlet om omsluttende rektangler og funksjonene som bruker en indeks uten å bli bedt om det.
- ST_DWithin — referansesiden for avstandstesten, med enhetene for hver type.
- Simple Feature Access – Part 1 — OGC-standarden bak geometritypene PostGIS lagrer.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er PostGIS?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/postgis/