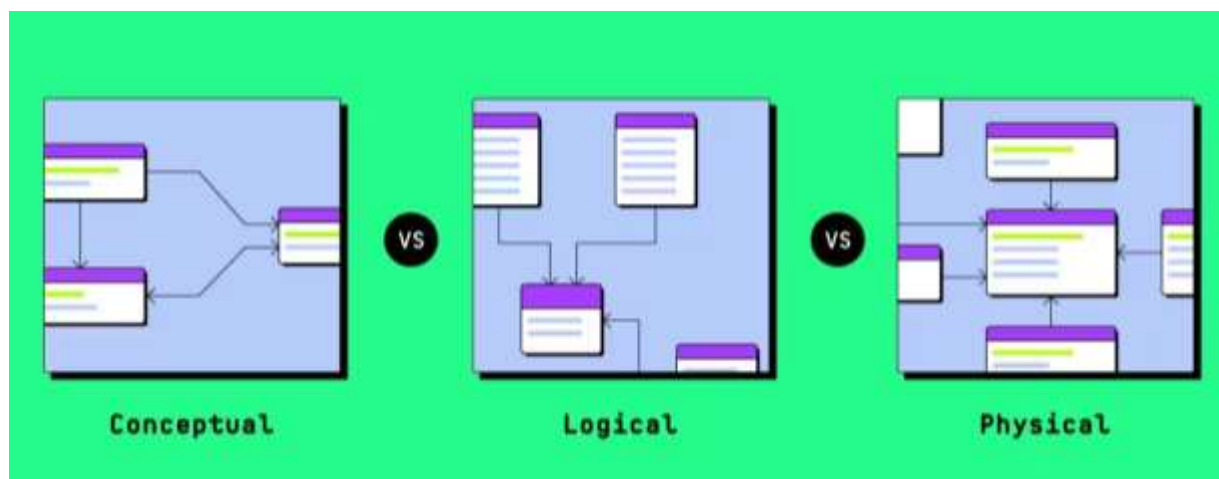


MODELE DE DATE CONCEPTUALE VS. LOGICE VS. FIZICE: CARE ESTE DIFERENȚA?



Dacă afacerea ta funcționează pe bază de date, atunci modul în care modelezi acele date contează. Nu doar pentru ingineri, ci pentru toți cei care iau decizii pe baza acestora.

Modelarea datelor nu înseamnă să desenezi casete și săgeți. Ci cum se aliniază echipele, cum se construiește încrederea și cum se scalează deciziile.

În continuare, vom analiza cele trei straturi ale modelării datelor:

conceptual, logic și fizic, și modul în care fiecare contribuie la:

1. **Comunicarea între echipele de business și cele de date**
2. **Date curate și consistente**
3. **Performanță la scară largă**
4. **Arhitectură pregătită pentru viitor**

Indiferent dacă schițezi un nou produs de date sau remediezi ceea ce există deja, înțelegerea acestor straturi te va ajuta să o faci mai bine și mai rapid.

MODELE DE DATE CONCEPTUALE VS. LOGICE VS. FIZICE: O COMPARAȚIE COMPLETĂ

Înainte de a analiza în detaliu fiecare model, iată un tabel comparativ rapid, urmat de o descriere detaliată a modului de interpretare a acestuia. Dacă sunteți nou în modelarea datelor, veți avea o idee despre ce acoperă fiecare model și cum se bazează unele pe altele.

Aspect	Modelul conceptual de date	Model logic de date	Modelul fizic de date
--------	----------------------------	---------------------	-----------------------

Scop	Definiți ce date sunt importante pentru afacere	Definiți modul în care datele ar trebui structurate logic	Definiți modul în care datele vor fi implementate într-o anumită bază de date
Public	Părți interesate din domeniul afacerilor, arhitecți de date	Analști de date, modelatori de date	Administratori de baze de date, ingineri
Focus	Entități și relații de afaceri	Atribute, chei, normalizare	Tabele, coloane, indexuri, constrângeri
Specific tehnologiei	Nu	Nu	Da
Include tipuri de date	Nu	Uneori (generalizat)	Da (specific platformei)
Include constrângeri	Nu	Da (constrângeri logice precum PK-uri, FK-uri)	Da (PK-uri, FK-uri, nu nule, unice etc.)
Normalizare	Nu se aplică	Da	Uneori (poate denormaliza pentru performanță)
Cazuri de utilizare	Alinierea părților interesate în ceea ce privește nevoile de date	Ghidarea proiectării bazelor de date, susținerea planificării analizelor	Stimulează implementarea efektivă, optimizarea performanței și scalarea
Exemplu de ieșire	Diagrama ER de nivel înalt	Diagramă ER detaliată cu câmpuri și relații	Scripturi SQL DDL, schemă de baze de date

IATĂ CUM SE INTERPRETEAZĂ DIFERENȚELE

În timp ce modelul conceptual cartografiază entitățile și relațiile de afaceri la nivel înalt, modelul logic adaugă mai multă structură, atribute, chei și normalizare, rămânând în același timp independent de tehnologie.

Între timp, modelul fizic este construit pentru implementare. Deși moștenește structura modelului logic, adaugă caracteristici specifice platformei, cum ar fi tipuri de date, indexuri și partiții, pentru a îmbunătăți performanța.

Fiecare se bazează pe precedentul:

- **Conceptualizarea** aliniază echipele de business și de date
- **Logica** oferă arhitecților un plan
- Implementare backend a ghidurilor **fizice**

Deși deserveșc publicuri diferite, cele trei modele funcționează cel mai bine împreună, formând o progresie lină de la înțelegerea afacerii la execuția tehnică. Acum, că am acoperit imaginea de ansamblu, haideți să analizăm fiecare tip de model de date mai detaliat.

Vom explora ce acoperă fiecare, cine îl utilizează și cum se încadrează în strategia dumneavoastră generală de date.

CE ESTE UN MODEL DE DATE CONCEPTUAL?

Un Model de Date Conceptual (MCD) este o reprezentare la nivel înalt a entităților principale din afacerea dvs. și a modului în care acestea se raportează una la cealaltă. Este conceput pentru a ajuta atât echipele tehnice, cât și pe cele non-tehnice să se alinieze asupra datelor importante, fără a intra în detalii despre structură sau stocare.

În timp ce modelele fizice se ocupă de tabele și tipuri de date, iar modelele logice definesc relații și chei, un model conceptual se referă la claritate. Este fundamentul pentru orice altceva.

Un model de date conceptual este:

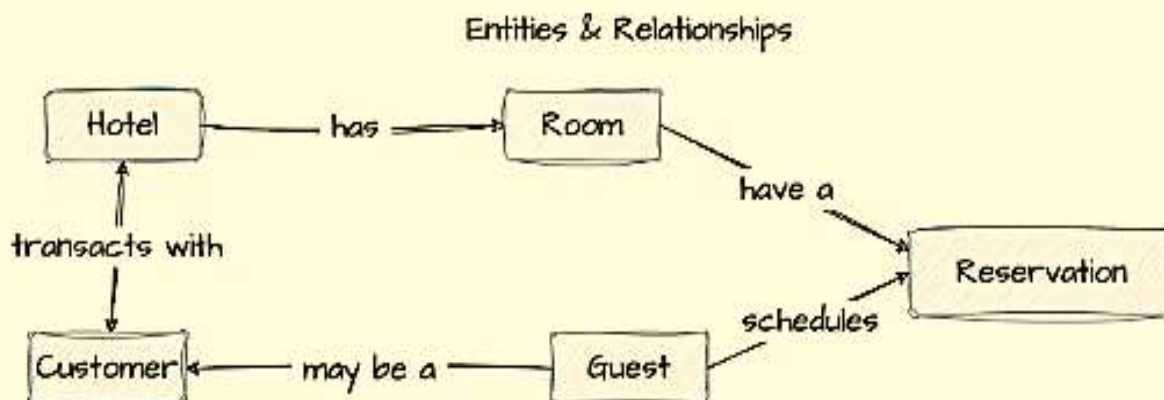
- **Platformă agnostică** : Nu se bazează pe nicio bază de date sau instrument specific
- **Orientat spre afaceri** : Folosește un limbaj care are sens pentru părțile interesate
- **Simplu și vizual** : De obicei, surprins într-o diagramă entitate-relație (DER)

Exemplu: Sistem de rezervări hoteliere

Iată o imagine simplificată a ceea ce ar putea include un model conceptual într-un sistem de rezervări hoteliere:

Entitate	Definiție	Relație
Hotel	O locație fizică care oferă camere oaspeților.	Are multe camere și deservește mulți clienți.
Cameră	O subunitate din cadrul unui hotel care poate fi rezervată.	Apartține unui singur hotel, poate fi rezervat de mai mulți oaspeți.
Oaspete	Cineva care stă la hotel.	Poate rezerva mai multe camere; poate fi și client.
Client	Oricine face o achiziție de la hotel, cum ar fi de la magazinul de cadouri.	Poate fi sau nu un oaspete.
Rezervare	O înregistrare a unei rezervări de cameră, identificată printr-un număr de confirmare.	Leagă un oaspete de o anumită cameră pentru un sejur.

Pentru a da viață acestui model, de obicei, reprezentați aceste entități și relațiile dintre ele într-o Diagramă Entitate-Relație (DER). Iată cum ar putea arăta aceasta:



Beneficiile unui model de date conceptual

Un model de date conceptual poate părea abstract, dar pune bazele esențiale. Iată de ce este important:

- **Înțelegere comună între echipe:** Reduce decalajul dintre părțile interesate de afaceri și echipele tehnice. Toată lumea pornește de la aceeași definiție a ceea ce contează.

- **Aliniere mai rapidă a proiectului:** Cu entități și relații clare mapate din timp, echipele petrec mai puțin timp dezbătând semantica în timpul implementării.
- **Comunicare mai simplă:** Modelul folosește termeni ușor de utilizat în mediul de afaceri, astfel încât nu este nevoie să traduceți jargonul tehnic pentru directori sau echipe interfuncționale.
- **Decizii de design scalabile:** Deoarece elimină detaliile de implementare, modelul conceptual vă permite să gândiți la scară mai largă, înainte de a fi constrânși de instrumente sau platforme.
- **O mai bună guvernanta a datelor:** Prin clarificarea entităților existente și a modului în care acestea se conectează, se pregătește terenul pentru convenții de denumire, proprietate și responsabilitate consecutive pe viitor.

CE ESTE UN MODEL LOGIC DE DATE?

Un Model Logic de Date (MLD) este o reprezentare structurată a datelor care prezintă entități, atribute și relații fără a face referire la nicio bază de date sau tehnologie specifică.

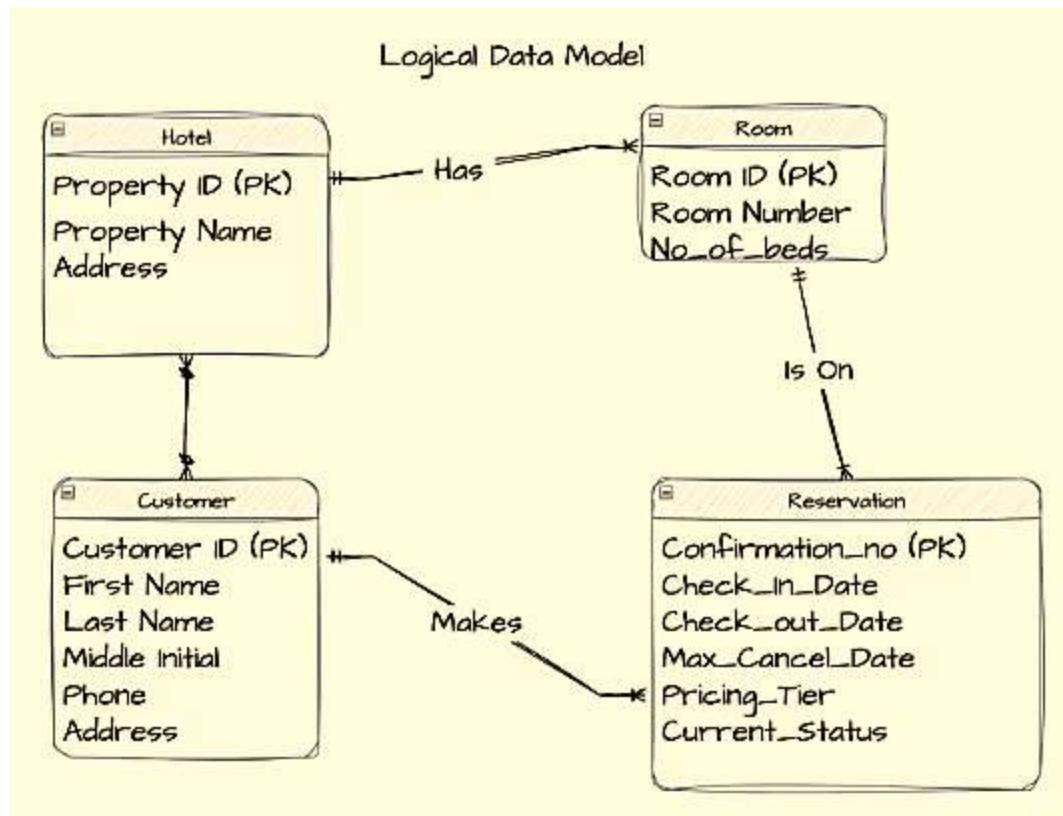
Se bazează pe modelul conceptual prin introducerea mai multor detalii, cum ar fi cheile primare și tipurile de date. Gândește-te la el ca la planul de lucru înainte de a începe să programezi.

Un model logic de date este:

- **Independent de platformă:** Încă nu este legat de nicio bază de date specifică
- **Mai detaliat:** Include câmpuri, chei și tipuri de date (la nivel înalt)
- **Normalizat :** Reduce redundanța pentru o integritate mai bună a datelor

Exemplu: Model logic pentru un sistem de rezervări hoteliere

Mai jos este un exemplu de diagramă ER care arată cum ar putea arăta un model logic de date pentru un caz de utilizare a rezervării hoteliere:



Beneficiile unui model logic de date

Modelele logice adaugă structură fără a te obliga să iei decizii tehnologice. Iată de ce reprezintă un pas esențial înainte de a trece la implementare:

- **Integritate mai puternică a datelor:** Cu chei definite și relații normalizate, modelul dvs. ajută la eliminarea duplicării și a inconsecvențelor, astfel încât datele dvs. rămân curate încă din prima zi.
- **Structură mai clară pentru dezvoltatori:** Modelele logice introduc suficiente detalii tehnice (cum ar fi atribute și chei) pentru a ghida implementarea, fără a intra în sintaxa specifică platformei.
- **Proiectare mai inteligentă a bazelor de date în fazele ulterioare:** Prin prelucrarea normalizării și a relațiilor din timp, reduceți problemele de reluare și performanță ulterioare în modelul fizic.
- **O mai bună aliniere cu regulile de business:** Nu doar proiectați pentru stocare, ci și cartografiați modul în care funcționează de fapt afacerea dvs. Această aliniere ajută la menținerea flexibilității și intuitivității sistemelor.
- **Fundație reutilizabilă pentru sisteme multiple:** Deoarece este independent de platformă, un model logic poate alimenta mai multe baze de date sau aplicații, susținând consecvența în întregul peisaj de date și

permițând analize self-service care se bazează pe date bine structurate și de încredere.

De la conceptual la logic: Cum să vă dezvoltați modelul

În modelul conceptual de date, entitățile și relațiile au fost toate definite. Următorul pas este utilizarea celor mai bune practici de modelare a datelor pentru a transforma acel concept de nivel înalt într-un model logic de date detaliat și normalizat.

Iată ce implică de obicei acest proces:

Pasul 1: Validați MCD -ul

Asigurați-vă că entitățile și relațiile din modelul conceptual sunt corecte și complete.

Pasul 2: Adăugați attribute

Identificați attributele suplimentare necesare pentru fiecare entitate.

Pasul 3: Identificați cheile candidate

Uită-te la attributele care identifică în mod unic fiecare entitate.

Pasul 4: Alegeți cheile primare

Lucrați cu părțile interesate pentru a alege cel mai bun candidat drept cheie primară, în mod ideal, un singur câmp.

Pasul 5: Normalizarea modelului

Folosește a treia formă normală (3FN) pentru a elimina redundanța. Sau, dacă proiectezi pentru analiză, ia în considerare modelarea dimensională pentru performanță.

Sfat de la profesionist: Dacă ceva nu se potrivește cu 3NF, încearcă să îl divizi într-o entitate nouă.

Pasul 6: Ajustați diagrama ER

Pe măsură ce normalizați, este posibil să fie nevoie să creați entități, attribute sau relații noi.

Pasul 7: Definiți relațiile

Validați relațiile dintre entități cu părțile interesate.

Sfat de la expert: Prea multe relații către o singură entitate? S-ar putea să aveți o eroare de design.

Pasul 8: Setări cardinalitatea

Decideți câte instanțe ale unei entități se leagă de alta (de exemplu, unu-la-mai-mult, mai-mult-la-mai-mult).

Pasul 9: Validarea modelului logic

Verificați-l în raport cu cerințele afacerii pentru a vă asigura că se potrivește cazului de utilizare din lumea reală.

Pasul 10: Iterare și rafinare

Modelarea datelor este rareori o soluție unică. Actualizați modelul pe măsură ce cerințele evoluează.

CE ESTE UN MODEL FIZIC DE DATE?

Un Model Fizic de Date (MFD) este un plan tehnic care mapează modelul logic de date la un sistem de baze de date specific, inclusiv depozite de date în cloud precum Snowflake sau BigQuery. Acesta prezintă exact modul în care datele vor fi stocate, indexate și accesate, până la tipul de date, lungimea coloanei și constrângeri.

Aici se întâlnesc designul cu realitatea.

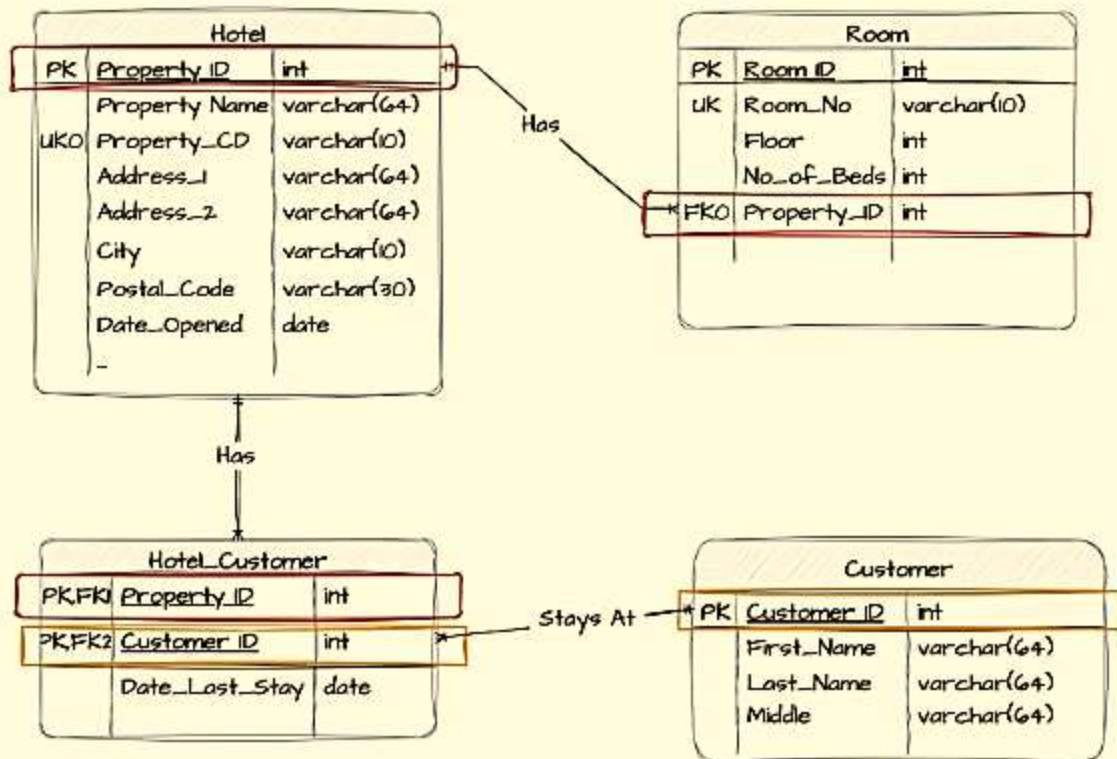
Un model fizic de date este:

- **Specific bazei de date** : Adaptat la o platformă specifică, cum ar fi Snowflake, BigQuery sau Postgres
- **Complet detaliat** : Include tipuri de date concrete, indexuri, constrângeri și formate de stocare
- **Orientat spre performanță** : Conceput pentru a optimiza viteza de interogare, eficiența stocării și scalabilitatea
-

Exemplu: Model fizic de date pentru un sistem de rezervări hoteliere

Mai jos este prezentat un model fizic de date derivat dintr-un sistem de rezervări hoteliere. Acesta include tipuri de câmpuri concrete, chei primare și externe și constrângeri relaționale.

Physical Data Model



Ce este diferit față de modelul logic:

- Cheile primare (PK) și externe (FK) sunt definite explicit
- Tipurile de câmpuri precum varchar(64) și int sunt specificate în funcție de cerințele bazei de date.
- Indexurile și constrângerile (de exemplu, UK pentru chei unice) sunt vizibile
- Relațiile dintre tabele sunt impuse prin integritate referențială reală

Beneficiile unui model fizic de date

Modelele fizice vă duc proiectarea până la execuție. Iată de ce sunt esențiale pentru construirea de sisteme performante și scalabile:

- **Gata pentru baza de date încă din prima zi:** Spre deosebire de modelele abstracte, un model de date fizic este gata de implementat în baza de date aleasă. Acesta definește tabele, tipuri de date, indexuri și reguli de stocare, astfel încât schema dvs. este pregătită pentru producție.

- **Interogări mai rapide, performanță mai bună:** Cu indexare, partiționare și nivelul potrivit de normalizare integrate, modelele fizice ajută sistemul să gestioneze interogări complexe și volume mari de date fără a încetini.
- **Se scalează odată cu afacerea dvs.:** Un strat fizic bine modelat susține creșterea, fie că asta înseamnă mai mulți utilizatori, mai multe date sau mai multe aplicații. Ați depus deja eforturile necesare pentru a susține scalarea fără o rearhitectură majoră.
- **Mai puține probleme legate de calitatea datelor:** Modelele fizice includ constrângeri care protejează datele, cum ar fi cheile primare, cheile externe și regulile la nivel de câmp. Aceasta înseamnă mai puține intrări greșite, mai puține erori și informații mai fiabile, toate acestea contribuind la o calitate mai bună a datelor în timp.

Sfat de la profesionist: Căutați instrumentele potrivite de modelare a datelor ? Asigurați-vă că acestea acceptă toate cele trei straturi, conceptual, logic și fizic, și că funcționează bine cu stiva dvs. de date modernă .

- **Plan clar pentru implementare:** De la crearea schemei până la alocarea spațiului de stocare, un model fizic ghidează administratorii de baze de date și inginerii prin procesul tehnic de construire. Toată lumea este aliniată și există mai puține incertitudini în timpul dezvoltării.

De la logic la fizic: Tranziția finală

Trecerea de la un Model Logic de Date (MLD) la un Model Fizic de Date (MFD) este rareori o sarcină unică. Este un proces iterativ care implică rafinarea structurii și încorporarea detaliilor specifice platformei pentru a susține modul în care datele vor exista și se vor comporta efectiv în baza de date.

Un design fizic solid începe cu un model logic solid, dar necesită și o înțelegere aprofundată a platformei de date cloud și a cazului de utilizare. Iată cum se desfășoară de obicei această tranziție:

Pasul 1: Selectați platforma de date

Decideți unde va fi amplasat modelul de date. Platforma aleasă va influența deciziile din etapele ulterioare și vă va permite să profitați de capacitățile specifice platformei.

Pasul 2: Conversia entităților logice în tabele fizice

Fiecare entitate din modelul logic este mapată la unul sau mai multe tabele fizice. Folosiți chei candidate din modelul logic pentru a defini cheia primară a fiecărui tabel.

Pasul 3: Definiți coloanele

Traduceți fiecare atribut într-o coloană și specificați tipul de date corespunzător (de exemplu, întreg, varchar, dată).

Pasul 4: Definiți relațiile

Stabiliți relații de cheie externă între tabelele părinte și copil prin crearea unei chei externe în tabelul copil care face referire la cheia primară a tabelului părinte.

Pasul 5: Verificarea normalizării (3NF)

Asigurați-vă că tabelele respectă a treia formă normală (3NF) pentru a minimiza redundanța și a menține integritatea datelor. Aceasta poate implica divizarea sau combinarea tabelelor pe baza principiilor de normalizare și a capacităților platformei.

Pasul 6: Definiți indexurile și partițiile

Creați indexuri și partiții bazate pe câmpuri frecvent interogate pentru a îmbunătăți performanța. Începeți simplu (KISS: păstrați-l simplu, inteligent) și optimizați iterativ pe baza comportamentului platformei și a volumului de lucru.

Pasul 7: Implementați constrângerile tabelului

Adăugați constrângeri precum chei primare, chei unice, verificări nule/non-nul și altele. Chiar dacă platforma de date nu impune toate constrângerile, definirea acestora poate fi benefică pentru instrumentele și aplicațiile din aval.

Sfat de la expert: Majoritatea platformelor de date în cloud nu impun fiecare constrângere, dar definirea acestora este în continuare importantă. Instrumentele de analiză și aplicațiile din aval le utilizează pentru a menține integritatea datelor și a îmbunătăți performanța.

Pasul 8: Adăugați elemente de programabilitate

În funcție de cazul de utilizare, implementați proceduri stocate, vizualizări, declanșatoare, fluxuri și activități pentru a sprijini automatizarea, analiza SQL și conductele de date automatizate .

Aceste elemente dau viață modelului tău fizic prin amplificarea operațiunilor în timp real, a transformărilor programate și a logicii de business personalizate.

Pasul 9: Validarea cu părțile interesate

Asigurați-vă că modelul fizic respectă cerințele de business descrise în modelul logic. Folosiți date de testare și exemple de interogări pentru a confirma acuratețea și performanța.

Un model logic bine structurat facilitează modelarea fizică, dar cele două nu sunt o predare simplă, ci sunt etape iterative, conectate. Scopul modelării fizice este de a transforma intenția afacerii într-un design de bază de date funcțional, performant și de încredere, care să reflecte atât modul în care organizația dumneavoastră gândește, cât și modul în care se comportă tehnologia.

Vrei să explorezi cum diferă modelele fizice de date în ceea ce privește structura și performanța? Consultă schema Star vs. schema Snowflake pentru a alege modelul potrivit pentru stratul tău de analiză.

De la modele la impuls

Deși modelarea datelor poate părea abstractă, impactul său este cu totul altceva. Atunci când este realizată corect, aceasta aliniază echipele, reduce reelaborarea lucrărilor și pregătește terenul pentru decizii rapide și sigure.

Modelul conceptual îi pune pe toți pe aceeași lungime de undă, în timp ce modelul logic adaugă structură, iar modelul fizic îl dă viață, fiecare strat construindu-se pe precedent.

În cele din urmă, modelul tău de date este strategia ta de date. Acesta modelează ceea ce urmărești, cât de repede obții răspunsuri și cât de bine scalezi.

Întrebări frecvente

1. Ce este modelarea datelor?

Modelarea datelor este procesul de creare a unei reprezentări vizuale sau conceptuale a datelor și a modului în care acestea circulă într-un sistem. Aceasta ajută la definirea structurii, relațiilor și regulilor care guvernează datele.

Gândește-te la el ca la planul datelor tale, acesta prezintă modul în care diferitele elemente de date se raportează între ele, astfel încât echipele să poată utiliza și analiza informațiile în mod consecvent.

2. Care sunt cele 5 tipuri de modele conceptuale?

Deși „modelul de date conceptual” se referă de obicei la un anumit tip de modelare a datelor, termenul „model conceptual” este mai larg și acoperă mai multe domenii. În proiectarea datelor și a sistemelor, iată cinci tipuri comune:

1. **Modelul entitate-relație (modelul ER):** Reprezintă datele folosind entități (lucruri) și relații dintre acestea. Destinat în proiectarea bazelor de date.
2. **Modelarea rolurilor obiectelor (ORM):** Se concentrează pe rolurile pe care obiectele le joacă în relații și este adesea utilizată în analiza sistemelor.
3. **Modelul semantic de date:** Pune accentul pe semnificație și context, folosind adesea ontologii și taxonomii pentru a defini relații și constrângeri.
4. **Model ierarhic:** Organizează datele într-o structură arborescentă, în care fiecare înregistrare are un singur părinte și potențial mai mulți copii.
5. **Model de rețea:** Similar modelului ierarhic, dar permite relații mai complexe între entități (mai-mulți-la-mai-mulți în loc de doar părinte-copil).

3. De ce am nevoie de diferite tipuri de modele de date? Nu pot trece direct la modelul fizic?

Omiterea modelelor conceptuale și logice poate duce la confuzie, echipe nealiniate și reproiectări costisitoare. Modelele conceptuale și logice îi ajută pe toți, de la părțile interesate din domeniu până la ingineri, să cadă de acord asupra datelor importante și a modului în care acestea ar trebui structurate înainte de a se apuca de implementarea tehnică.

4. Ce se întâmplă dacă cerințele afacerii mele se modifică după ce construiesc modelele de date?

Modelarea datelor este un proces iterativ. Va trebui să revizuiți și să ajustați modelele pe măsură ce afacerea dvs. evoluează. De aceea, pornirea cu modele conceptuale și logice ajută la actualizarea lor mai ușoară și mai ieftină înainte de a fi implementat modelul fizic.

5. Poate fi utilizat un model logic cu mai multe baze de date fizice?

Da. Deoarece modelele logice sunt independente de platformă, puteți utiliza același design logic ca plan pentru a crea mai multe modele fizice pentru diferite sisteme de baze de date, ceea ce ajută la menținerea consecvenței în mediul de date.

6. Care sunt câteva greșeli frecvente de evitat la modelarea datelor?

Printre capcanele se numără începerea prea devreme a designului fizic, ignorarea contribuțiilor din partea companiei, complicarea excesivă a modelelor cu entități sau relații inutile și nevalidarea regulată a modelelor cu părțile interesate. De asemenea, omiterea normalizării poate duce la date dezordonate și redundante.