

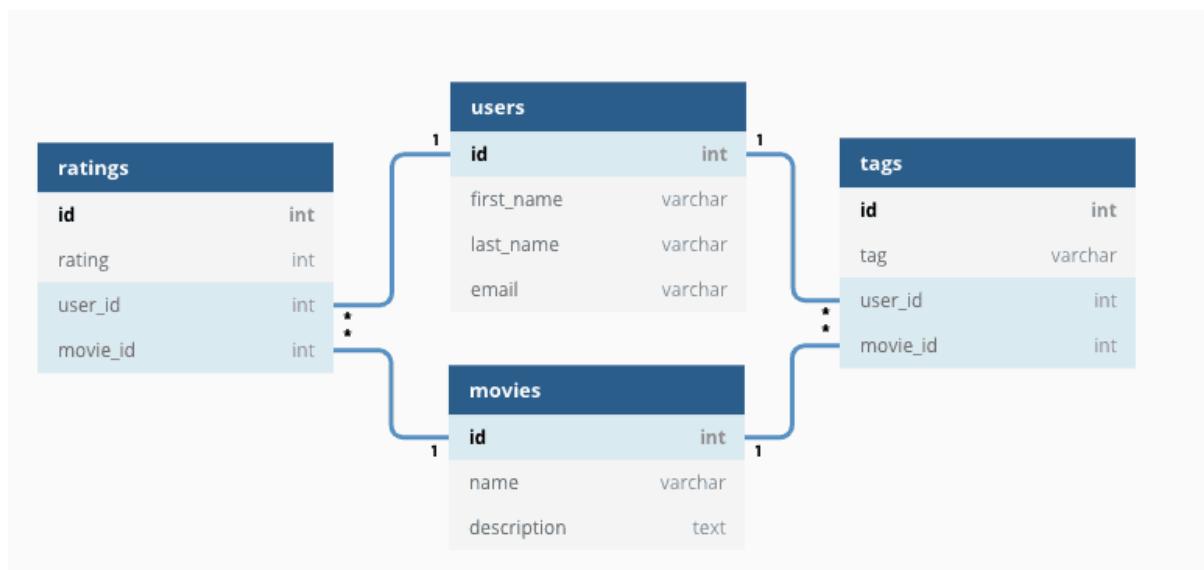
MA'LUMOTLAR BAZASINI LOYIHALASH

Ushbu maqolada web dastur yaratish jaryonida turli ehtiyojlarga mos keladigan ma'lumotlar bazasini yaratishni o'rghanamiz. Ma'lumotlar bazasini yaratish asoslari, shuningdek optimal natijalar uchun uni takomillashtirish usullari haqida ma'lumot beramiz.

Ma'lumotlar bazasi nima?

Ma'lumotlar bazasi (MB) bu - ma'lumotlarni o`qish, saqlash, qayta ishslash va qidirish uchun mo'ljallangan markaziy ma'lumotlar ombori. Turli dasturlar yaratish ekanmiz, dastur ishlashiga oid bo'lgan ma'lumotlarni u yoki bu ko'rinishda saqlashimiz kerak. Misol uchun dastur foydalanuvchilari va ularning shaxsiy ma'lumotlari, rasmlari, postlar, mahsulotlar va hokazo.

Bu ma'lumotlarni qay ko'rinishda saqlash, turli ma'lumotlar o'rtasidagi munosabatlari (foydalanuvchi-post, mijoz-mahsulot), bu ma'lumotlarga murojaat qilish va unumli foydalanish uchun MBni to'g'ri loyihalashda juda muhim ahamiyatga ega.



Ma'lumotlar bazasini loyihalash jarayoni

Yaxshi tuzilgan ma'lumotlar bazasi:

- Ortiqcha (takrorlanuvchi) ma'lumotlarni bartaraf qilish orqali xotirani tejaydi.
- Ma'lumotlarning aniqligi va yaxlitligini saqlaydi.
- Ma'lumotlarga foydali usullar bilan kirishni ta'minlaydi.

Samarali va foydali MBni loyihalashtirish quyidagi jarayonlarni o'z ichiga oladi:

1. Talablarni tahlil qilish yoki ma'lumotlar bazasining maqsadini aniqlash
2. Ma'lumotlarni jadvallarga ajratish
3. Asosiy kalitlarni (primary key) ko'rsatish va munosabatlarni tahlil qilish
4. Jadvallarni standartlashtirish uchun normallashtirish

Keling, har bir qadamni batafsil ko'rib chiqaylik. Ma'lumotlar bazasinining turli modellari juda ham ko'p, biz esa ushbu qo'llanmada ko'pchiilik tomonidan ishlataladigan relyatsion ma'lumotlar bazasi modelidan (**relational model**) foydalanamiz.

Dastavval Relational Model haqida to'htalaylik.

Relational Database Model

Eng keng tarqalgan model, relyatsion model ,ma'lumotlarni jadvallarga ajratadi. Har bir jadval ustunlar va qatorlardan iborat. Har bir ustunda biror obyektga oid atributlar keltirilgan, masalan narx, pochta indeksi yoki tug'ilgan sana va hokazo. Bu atributlar birgalikda **domen** deb ataladi.

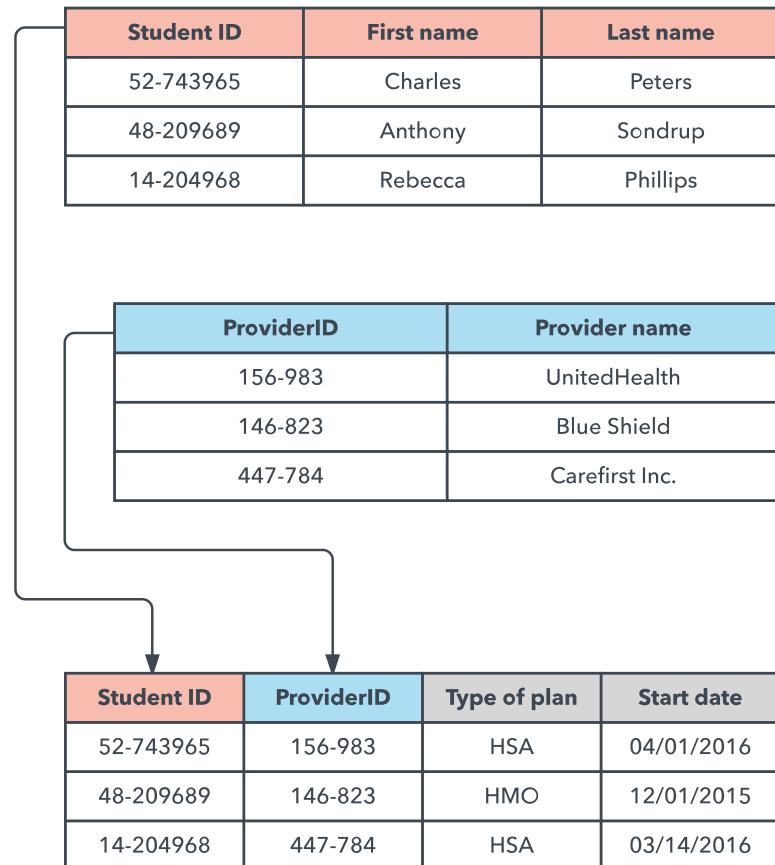
Quyida biz Foydalanuvchilar haqida ma'lumot saqlash uchun **Foydalanuvchilar** jadvalini yaratdik. Jadvaladagi **ID**, **Ism**, **Familiya**, **Telefon** foydalanuvchining **attributlari** hisoblanadi. Birgalikda esa attributlar **domen** deb ataladi.

Foydalanuvchilar

Aa	ID	Ism	Familiya	Telefon
	<u>23215462</u>	Anvar	Narzullaev	<u>99890123456</u>
	<u>45613246</u>	Olim	Qodirov	<u>98891123456</u>
	<u>4642457</u>	Botir	Rasulov	<u>99897123456</u>

MB bir nechta jadvallardan iborat bo'lishi tabiiy. Misol uchun **Mahsulotlar** haqida ma'lumotlarni saqlash uchun alohida jadval, **Xaridlar** uchun alohida jadval, **Mijozlar** uchun alhoida jadval va hokazo. Biror jadvalni boshqa jadvlaga bog'lash uchun jadval attributlaridan biri (yoki attriburtlar birikmasi) **primary (asosiy) kalit** etib tayinlanadi. Boshqa jadvallar esa bu attributga **foreign key** deb murojaat qiladi.

Relational Model jadvallar o'rtaqidagi munosabatlarni ham tartibga soladi. Masalan mijozlar va ular sotib olgan mahsulotlar jadvali, yoki talabalar va ular o'qiyotgan fanlar alohida jadvallarda saqlansada, ular o'rtaqidagi munosabat relational model yordamida ko'rsatiladi. Bunday munosabatlar **one-to-one** (birga-bir), **one-to-many** (birga-ko'p) yoki va **many-to-many** (ko'pgal-ko'p) ko'rinishida bo'lishi mumkin. Munosabatlar haqida birozdan keyin alohida to'htalamiz.



Demak yuqoridagilarning barchasi Relational Ma'lumotlar bazasining sifatlari ekan.

Keling endi bevostia loyihalash jarayoniga to'htalamiz.

1. Talablarni tahlil qilish: ma'lumotlar bazasining maqsadini aniqlash

Dasturchi sifatida yaratilayotgan dasturiy ta'minot, unda saqlanadigan ma'lumotlar va ularga qo'yilgan talablarni tushunish ma'lumotlar bazasini loyihlashda ishingizni bir muncha osonlashtiradi.

Ma'lumotlar bazasiga har tomonlama qarashni unutmang. Masalan, onlayn do'kon uchun baza yaratganda, bazaga mijoz sifatida, yoki sotuvchi sifatida qarashingiz mumkin va bu ikkisi uchun bir hil ma'lumotlar turli ko'rinishda ko'rsatilishi kerakligini hisobga oling.

Talabalar va maqsdalarni aniqlash uchuh quyidagi qadamlarni bajarish mumkin:

- Uni ishlatajigan odamlardan intervyu oling
- Hisob-fakturalar, ish jadvallari, so'rovnomalar kabi biznes hujjatlarni tahlil qiling
- Mavjud ma'lumotlar tizimlarini o'rganing (shu jumladan jismoniy va raqamli fayllar)

Navbat ma'lumotlar bazasiga kiritilishi kerak bo'lgan barcha ma'lumotlarni yig'ib, ularni turlarga bo'lishga: shaxslar, joylar, buyumlar, hodisalar va hokazo. Har bir turdag'i ma'lumotlarning attributlarini ham aniqlab oling:

Xaridorlar

- Ism
- Manzil
- Shahar, tuman, viloyat
- E-pochta manzili
- Telefon raqami

Mahsulotlar

- Nomi
- Narxi
- Zaxiradagi miqdori
- Buyurtma miqdori

Buyurtmalar

- Buyurtma ID
- Buyurtmani qabul qilgan sotuvchi
- Sana
- Mahsulotlar
- Miqdor
- Narx
- Jami

Keyinchalik bu ma'lumotlar bevosita MB jadvallarini shakllantirishda ishlatiladi. Ma'lumotlarni imkon qadar mayda bo'laklarga bo'ling. Misol uchun, manzilni shahar, tuman, viloyat qilib ajratsangiz, kelajakda MBdagi foydalanuvhchilarni tuman, shahar yoki viloyat bo'licha ajratib olishingiz qulay bo'ladi.



Yana bir muhim jihat, bitta ma'lumotni bir nechta jadvalda qayta saqlashdan ogoh bo'ling.

Tasavvur qiling, sizda **foydalanuvchilar** va **mijozlar** degan jadvallar mavjud va siz biror shaxs haqidagi ma'lumotlarni ikkallasida ham saqlayabsiz. Endi agar bu shaxsnинг ma'lumotlari bitta jadvalda yangilansa (masalan telefon raqam), siz ikkinchi jadvalni ham yangilashingizga to'g'ri keladi. Bu esa ortiqcha ish. Buning oldini olish uchun **foydalanuvchilar** uchun bitta jadval yaratib, bu foydalanuvchi mijoz yoki yo'qligini **attribut** sifdatida ko'rsatish mumkin.



Ma'lumotlar bazasi qaysi turdagи ma'lumotlarni o'z ichiga olishi, bu ma'lumotlar qayerdan olinishi va ulardan qanday foydalanilishini o'rganganingizdan so'ng, siz haqiqiy ma'lumotlar bazasini rejorashtirishga tayyor bo'lasiz.

2. Ma'lumotlar Bazasining tuzilishini reajalash

Navbat ma'lumotlar bazasining tashq ko'rinishini shakllantirishga. Yuqirda ko'rganimizdek, Relational MB jadvallardan iborat bo'ladi. Har bir jadval esa ustunlar (attributlar) va qatorlardan tashkil topadi. Har bir qator ingliz tilida record deb ataladi va bitta buyum (shaxs) haqida ma'lumot saqlaydi.

Talabalar

#	ID	Aa Ismi	Guruhi	Fakultet
456234	<u>Alijon Valiyev</u>	B120	Matematika	
456454	<u>Valijon Aliyev</u>	B120	Matematika	
123456	<u>Hasan Husanov</u>	B120	Matematika	

Jadvalning har bir qatoridagi ma'lumotlarni bir hilda saqlash uchun, ularning turini ham tayinlab ketish maqsadga muvofiq bo'ladi. Masalan **Ism** faqat matn, **ID** esa faqat raqam.

Umuman olganda ma'lumotlarni saqlashning quyidagi turlari mavjud:

- CHAR - doimiy uzunlikdagi matnlar uchun
- VARCHAR - o'zgaruvchan uzunlikdagi matnlar uchun
- TEXT - katta matnlar uchun
- INT - manfiy yoki musbat butun sonlar
- FLOAT, DOUBLE - manfiy yoki musbat o'nlik sonlar
- BLOB - ikkilik sanoq tizimidagi ma'lumotlar

Shuningdek aksar ma'lumotlar omborlarini boshqaruv tizimlari har bir qator (record) uchun noyob ID raqamini ham qo'shib ketadi.

Bu qadamda ma'lumotlar bazasini ko'z oldingizga keltirish uchun jadvallarni maxsus dastur yordamida yoki oddiy qog'ozda to'rtburchak ko'rinishida chizishingiz mumkin. Bunda jadvalni to'liq emas, faqatgina unga tegishli attributlarni ko'rsatish kifoya.

Student
Student ID
Birth date
Grade level

Nihoyat, jadval attributlaridan birini **birlmachi kalit (primary key)** sifatida tanlaysiz. Bu attribut noyob bo'lib, faqatgina yagona qator (record)ga tegishli bo'lishi kerak. Misol uchun shaxsning ID yoki passport raqami noyob kalit bo'lishi mumkin. Bi faqatgina shu yagona kalit yordamida ro'yxatdan bizga kerakli qator (shaxs, mahsulot) ni ajrata olishimiz kerak.

A **Birlmachi kalit qilib tanlangan attribut nafaqat noyob va o'zgarmas, balki doimo mavjud bo'lishi ham shart. Ya'ni bazaga ma'lumot qo'shilganda bu qiymatni kiritmay tashlab o'tish mumkin emas.**

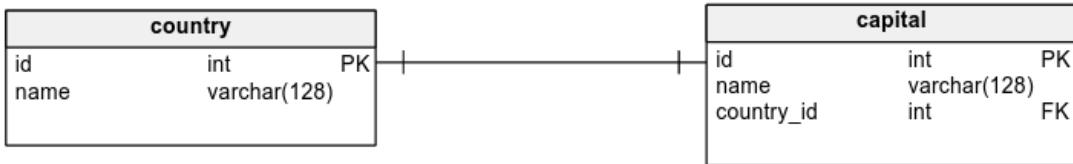
3. Jadvallar o'rtaida munosabat

Ma'lumotlar bazasi bir necha turdag'i jadvallardan iborat, navbat bu jadvallar o'rtaidagi munosabatlarni tahlil qilishga. Masalan, mijozlar va mahsulotlar, talabalar va darslar, o'qituvchilar va o'quvchilar o'zaro qanday bog'langan? Ular o'rtaidagi munosabat qay ko'rinishda bo'lishi kerak? Navbat shu savollarga javob berishda va munosabatlarni shakllantirishda.

Jadvallar o'rtaidagi munosabatlar quyidagi 3 ta ko'rinishda bo'lishi mumkin:

One-to-one (birga-bir)

Jadvaldagi bitta qator (record) boshqa jadvaldagi bitta qatorga bog'langan bo'ladi. Bunday munosabat kamdan-kam uchraydi va umuman olganda bunday jadvallarni jamlab yuborish tavsiya qilinadi. Misol uchun quyida birinchi jadval mamlakat (**country**) ikkinchi jadval esa poytaxt (**capital**) haqida ma'lumotlarni saqlaydi. Bu ikki jadval esa birga-bir munosabat bilan bog'langan. Ya'ni, bitta davlatning bitta poytaxt bo'lishi mumkin, bitta poytaxt esa boshqa davlatga poytaxt bo'la olmaydi.

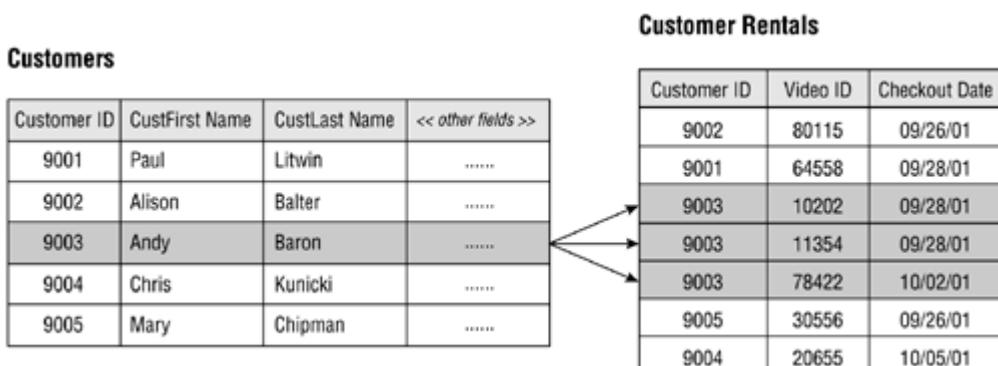


Yuqorida aytganimizdek, bunday holatlarda poytxat nomini davlatning attributi sifatida saqlagan afzal.

country		
		PK
id	int	
name	varchar(128)	
capital		varchar(128)

One-to-many (birga-ko'p)

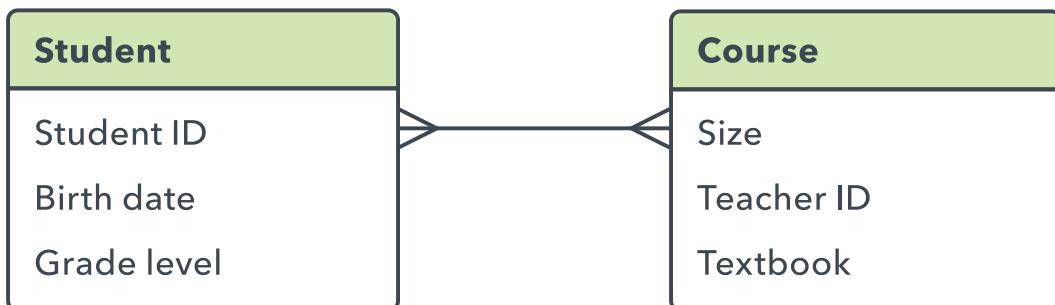
Bunda bir jadvaldagi bitta qator (record) boshqa jadvaldagi bir nechta qatorlarga bog'langan bo'ladi. Masalan bitta mijoz bir nechta mahsulotlar xarid qilishi, bitta muallif bir nechta kitoblar chop etishi mumkin va hokazo. Bu munosabat one-to-many deb ataladi.



Bunday ma'lumotlar bazasini yaratishda, birinchi jadvaldagi birlmachi kalit, ikkinchi jadvalning attributi sifatida qo'shiladi. Yuqoridagi misolda **Customers** (mijozalr) jadvalidagi **Customer ID** birlmachi kalit hisoblanadi va bu kalit **Customer Rentals** jadvalida ham mavjud. Mijozning ID raqamiga qarab, mijoz aynana qaysi mahsulotlarni olganini ko'rishimiz mumkin. Masalan, **Andy** ismli mijozning ID raqami **9003** va u kulrang bilan ajratilgan mahsulotlarni sotib olgan.

Many-to-many (ko'pga-ko'p)

Jadvaldagi bir nechta qatorlar, boshqa jadvladagi bir nechta qatorlar bilan bog'lanishi **many-to-many** munosabati deyiladi. Masalan, bitta talaba bir nechta fanlarga yozilishi mumkin, shu bilan birga bitta fanga bir nechta talaba yozilishi ham mumkin.



Afsuski, bunday munosabatni bevosita dasturlash murakkab (yoki ilojsiz). Shuning uchun bu ikki jadval o'rtaida bog'lovchi jadval yaratish maqsadga muvofiq bo'ladi.

Misol uchun bizda **mijozlar** va **mahsulotlar** jadvali bor. Mijozlar istalgancha mahsulot sotib olishi mumkin, mahsulotlar esa istalgancha mijozga sotilishi mumkin. Ya'ni bu ikkisi many-to-many munosabatda. MB shakllantirish uchun biz **mijzolar** va **mahsulotlar** ni bog'lovchi, yangi **xaridlar** (yoki **savat**) jadvalini yaratamiz. Mijozlar ham mahsulotlar ham xaridlar jadvaliga one-to-many munosabatda bo'ladi.

Majburiy yoki yo'q?

Munosabatlarni shakllantirishda qatorlardan (record) qaysi biri birlamchi ekanini aniqlashtirib olish ham muhim. Ya'ni bir jadvaldagi qator mavjud bo'lishi uchun ikkinchi jadvaldagi qator bo'lishi majburiy mi yoki yo'q. Misol uchun mualliflar va kitoblar jadvalida, muallif Kitobsiz bo'lishi mumkin, lekin kitob muallifsiz bo'lmaydi. Ya'ni kitoblar jadvalida kitob mavjud bo'lishi uchun mualliflar jadvalida kitobning muallifi mavjud bo'lishi shart. Aksincha, mualliflar jadvalida biror muallifning mavjudligi, kitoblar jadvalida shu muallifning kitoblari bo'lishini taqozo qilmaydi (balki muallif hali kitob yozmagan, yoki kitobi bosmadan chiqmagan, yoki onlayn do'konda bu muallifning kitoblari mavjud emas).

Rekursiv munosabat

Ba'zida jadval qaytaib o'ziga murojaat qilishi mumkin. Bu rekursiv munosabat deyiladi. Misol uchun korxona ishchilari jadvalida boshliq degan attribut bo'lsa, bu attribut shu jadvaldagi boshqa odamga (boshliqqa) ishora qiladi.

Ortiqcha munosabat

Ba'zida jadvallardagi qatorlar (record) bir-biri bilan birdan ko'p munosabatda bo'ladi. Masalan, talabalar o'qituvchilar bilan bo'glangan, o'qituvchilar fanlarga bog'langan, talabalar ham fanga bog'langan. Ya'ni talabalar o'qituvchilarga ikki marta bog'lanib qoldi, biri tog'ridan-to'g'ri, ikkinchisi fan orqali. Bunday holatda munosabatlardan biridan voz kechish MBga ta'sir qilmaydi va ishimizni osonlashtiradi. Yuqoridagi misolda, talabalar ham o'qituvchilar ham biror fanga bog'lanishi kifoya qiladi.

Davomi bor...