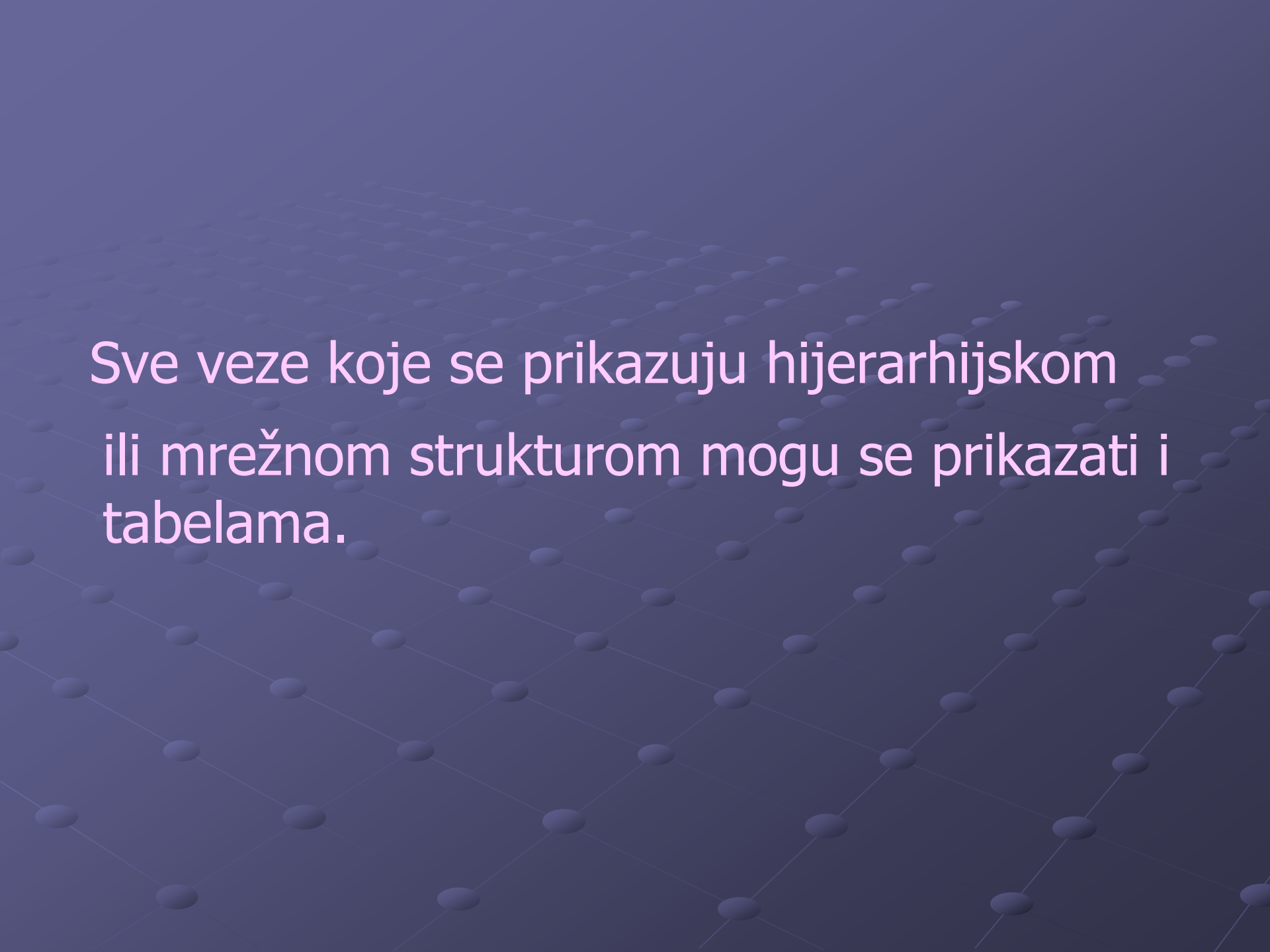
The background is a dark blue gradient with a subtle, repeating pattern of light blue dots connected by thin lines, creating a perspective effect that recedes into the distance.

RELACIONE BAZE PODATAKA

The background is a dark blue gradient with a subtle, repeating pattern of light blue dots connected by thin lines, creating a grid-like or network structure that recedes into the distance.

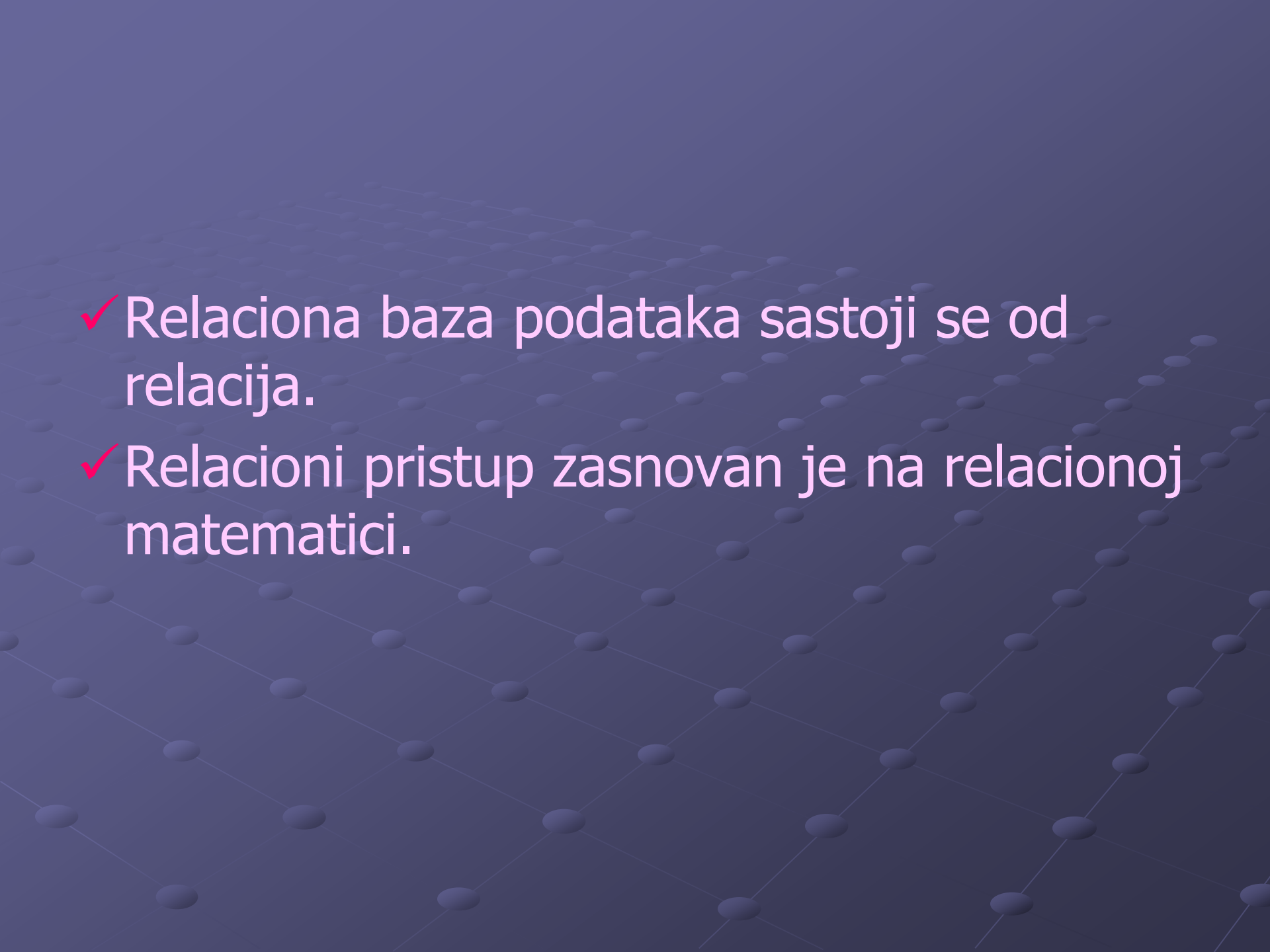
Sve veze koje se prikazuju hijerarhijskom ili mrežnom strukturom mogu se prikazati i tabelama.

Generalno uloga tabele u relacionoj bazi podataka je ta, da tabela predstavlja svrstavanje podataka u vrste. Sledeća tabela daje sažet prikaz sličnosti između koncepta baze podataka i njihovih tabelarnih zapisa:

TERMINOLOGIJA RELACIONE BAZE PODATAKA	TABELARNA TERMINOLOGIJA
<i>relaciona baza podataka</i>	skup tabela
<i>relacija</i>	jedna tabela
<i>atribut (karakteristika)</i>	zaglavlje kolone u tabeli
<i>slog</i>	vrsta od podataka u tabeli
<i>mera veličine skupa određenih objekata</i>	broj vrsta u tabeli
<i>stepen</i>	broj kolona u tabeli
<i>domen</i>	postavka važećih vrednosti za podatke u koloni

Tabele imaju uglavnom sledeće osobine:

- ✓ ne postoje grupe koje se ponavljaju i svaka rubrika tabele predstavlja jedan podatak,
- ✓ u jednoj koloni postoji samo jedna vrsta podataka,
- ✓ redovi se razlikuju među sobom,
- ✓ kolone imaju svoj naziv.

- 
- ✓ Relaciona baza podataka sastoji se od relacija.
 - ✓ Relacioni pristup zasnovan je na relacionoj matematici.



✓ Skup vrednosti jednog podatka odnosno kolone naziva se domen.

✓ Relacija je jednoznačno identifikovana ključem relacije. Svaka relacija mora imati ključ.

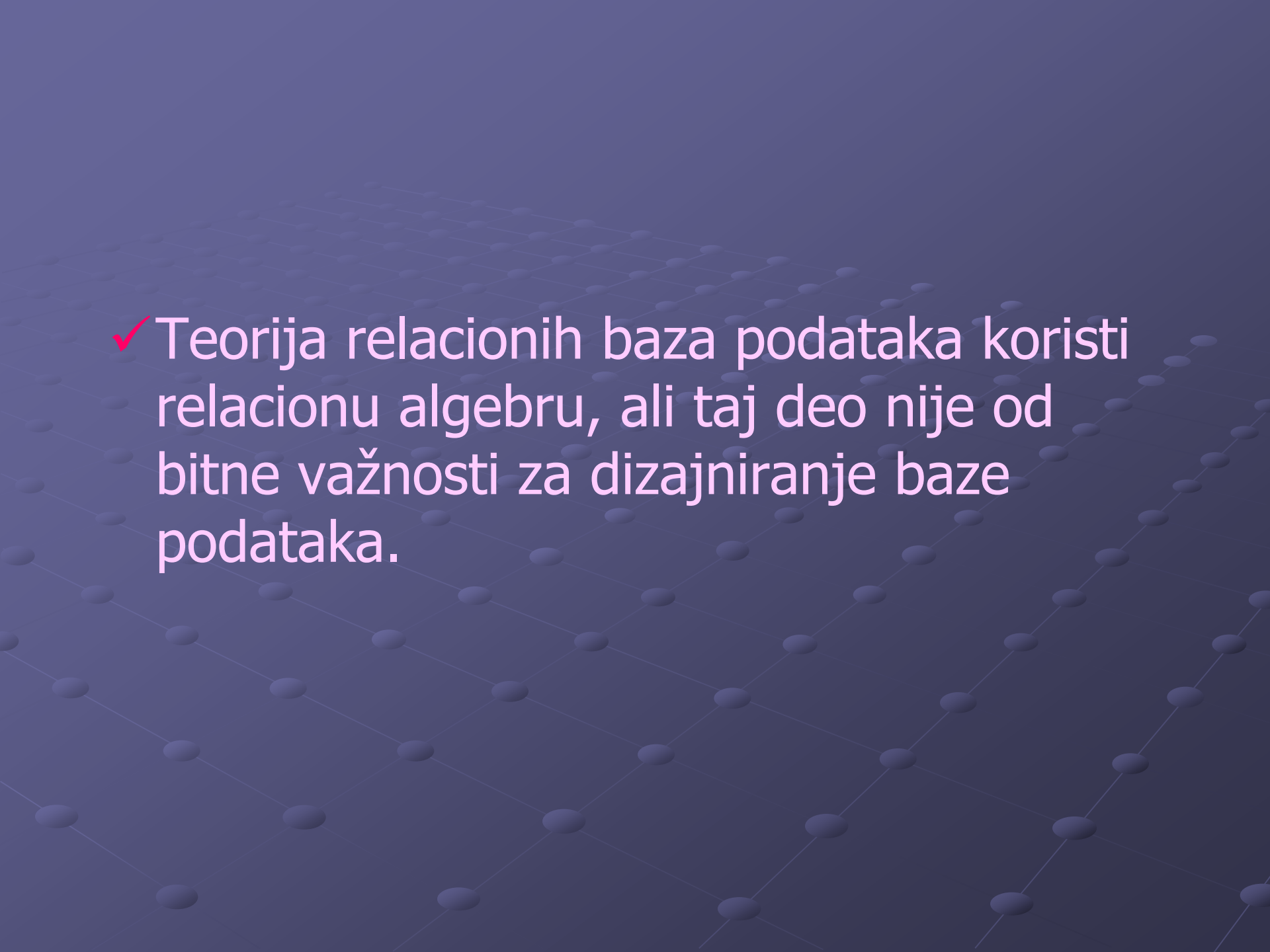
- ✓ S obzirom na to da su baze podataka okrenute širokom krugu korisnika, različiti korisnici baze podataka videće različite skupove podataka i različite relacije između njih.

Ukoliko su potrebne informacije iz više tabela i ukoliko te tabele imaju neki zajednički element podataka, model dolazi do punog izražaja, jer omogućava povezivanje tabela ili delova tabela.

Primer relacije mogao bi se prikazati kao:

UČENIK (lični broj, ime, odeljenje)

pri čemu je podatak ispred zagrade naziv relacije (tabele), podaci u zagradi su nazivi domena (polja), a podvučeni podaci su ključevi za identifikaciju redova.

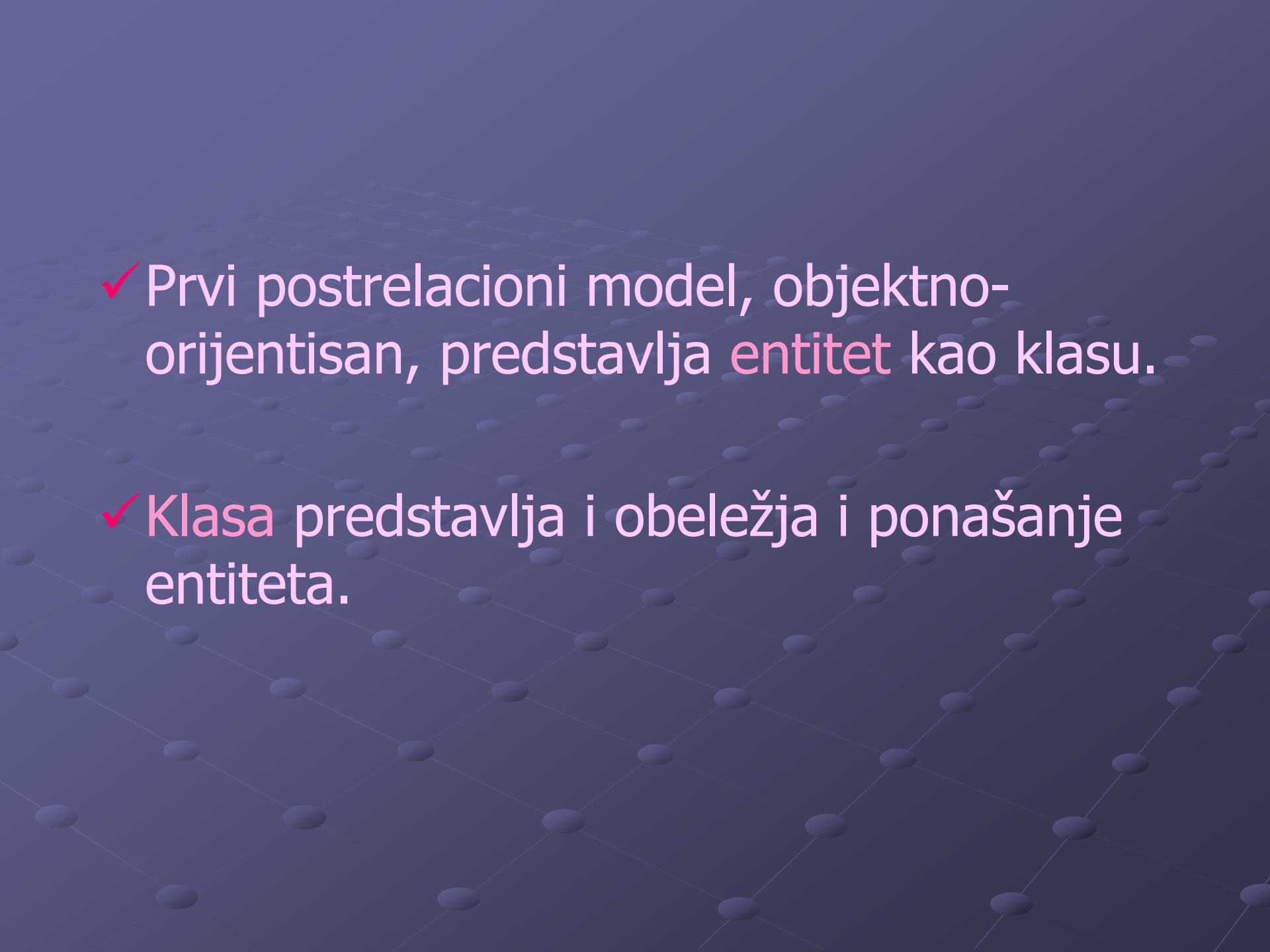
A 3D grid of blue spheres on a dark blue background. The spheres are arranged in a perspective view, creating a sense of depth. The background is a solid dark blue color.

✓ Teorija relacionih baza podataka koristi relacionu algebru, ali taj deo nije od bitne važnosti za dizajniranje baze podataka.

- ✓ **Relaciona algebra** je potrebna za precizan opis operacija koje se izvode nad podacima u relacionoj bazi podataka čime se postiže veza između opisa podataka u korisnikovom pogledu i opisa podataka u konceptualnom modelu podataka konkretne baze podataka.

Objektno-orijentisan model baza podataka

- ✓ Iako relacioni model dominira, tzv. postrelacioni modeli treba da omoguće fleksibilnije prikazivanje podataka.

- 
- ✓ Prvi postrelacioni model, objektno-orijentisan, predstavlja **entitet** kao klasu.
 - ✓ **Klasa** predstavlja i obeležja i ponašanje entiteta.

Bilo koji sistem se može posmatrati kao skup međusobno povezanih objekata. Pod objektima u nekom sistemu se podrazumevaju fizički objekti, koncepti, apstrakcije, bilo šta što ima jasne granice i jasno značenje, što se jasno razlikuje od drugih objekata u sistemu. Objektni model polazi od ovakve definicije sistema.

U objektnom modelu definišu se sledeći koncepti:

✓ **Objekat** i **literal** su osnovni primitivni koncepti objektnog modela. Pod objektom se podrazumeva entitet koji je sposoban da čuva svoja stanja i koji stavlja okolini na raspolaganje skup operacija preko kojih se ta stanja prikazuju ili menjanju.

Literal je u osnovi vrednost, podatak koji se koristi u modelu. Literali mogu biti prosti, složeni ili kolekcije. Pojam literala je ekvivalentan pojmu nepromenljivog objekta koji se koristi u nekim ranijim objektnim modelima i programskim jezicima. Posebno je potrebno naglasiti razliku između objekta i literala: *objekat ima jedinstveni identifikator, a literal nema.*

✓ Objekti i literali se kategorizuju u tipove. Svi objekti, odnosno literali istog tipa imaju jedinstven skup **stanja (osobina)** i jedinstveno ponašanje (odnosno isti skup operacija). Konkretni objekat se ponekad naziva **pojavljivanje (instanca)** datog tipa.

✓ Stanje objekta se predstavlja vrednostima njegovih osobina. Pod osobinama objekta se podrazumevaju atributi objekta i njegove veze sa drugim objektima u sistemu. Osobine objekta se menjaju u vremenu.

✓ **Ponašanje objekta** se opisuje preko skupa operacija koje on izvršava ili se nad njim izvršavaju. Svaka operacija ima kao implicitni argument objekat kome je pridružena. Pored toga, operacija može da ima listu ulaznih i izlaznih parametara definisanih tipova, a može i da vrati tipizovan rezultat.

✓ Baza podataka skladišti objekte i stavlja ih na korišćenje većem broju korisnika, odnosno aplikacija. Baza podataka se opisuje preko svoje šeme koja se specifikuje preko ODL-a (*Object Definition Language* – Objektni specifikacioni jezik). U šemi se definišu tipovi objekata čija se pojavljivanja čuvaju u bazi.

The background is a dark blue gradient with a subtle, repeating pattern of light blue dots connected by thin lines, creating a grid-like or network structure that recedes into the distance.

✓ Objektno-orijentisani model održava
relacije kroz logički sadržaj.

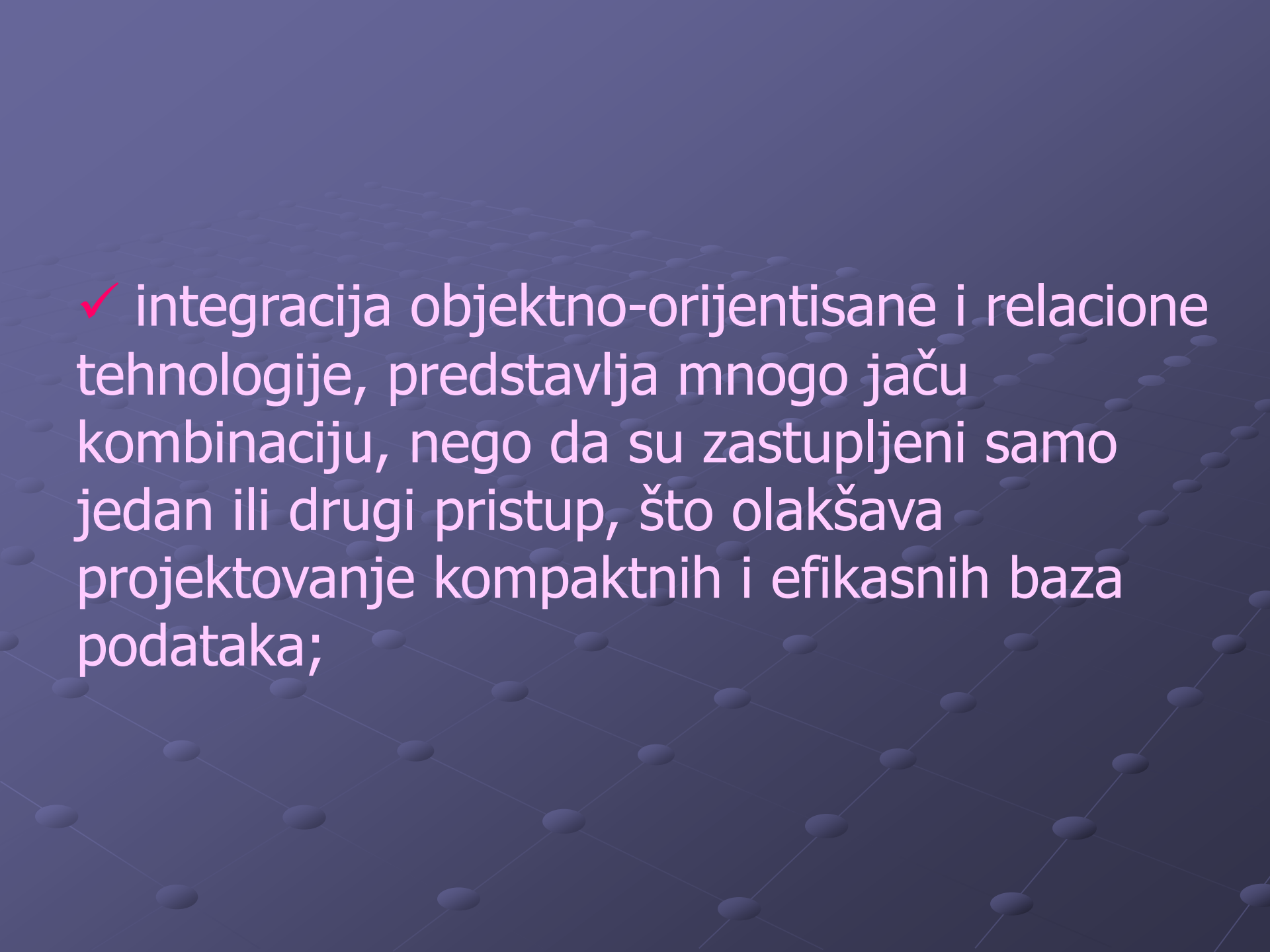
- ✓ Drugi postrelacioni model baza podataka je **deduktivni**, poznatiji kao izvedeni model.
- ✓ Ovaj model čuva što je moguće manje podataka, što kompenzuje pravilom da kombinovanjem stvara nove podatke po potrebi.

Objektno-relacioni model baza podataka

Postoje razlozi zbog čega se smatra da će u budućnosti prevladati korišćenje objektno-relacionog pristupa:

- ✓ objektno-relacione baze podataka, progresivno su kompatibilne sa trenutnim korisničkim relacionim bazama podataka;

✓ mogućnost postavljanja trenutnih upita.
Prethodne objektno-orijentisane baze podataka nisu podržavale standardne mogućnosti upita SQL baza podataka, odnosno problem je bio u primeni standardnih SQL alata u objektno-orijentisanim bazama podataka;



✓ integracija objektno-orijentisane i relacione tehnologije, predstavlja mnogo jaču kombinaciju, nego da su zastupljeni samo jedan ili drugi pristup, što olakšava projektovanje kompaktnih i efikasnih baza podataka;

Osnovne karakteristike objektno-relacione tehnologije mogu se definisati kroz:

- ✓ Objekte;
- ✓ Klase;
- ✓ Učaurenja;
- ✓ Mogućnosti proširenja;
- ✓ Nasleđivanje;
- ✓ Polimorfizam.

Objekti su softverski prikazi entiteta iz stvarnog sveta. Sastoje se iz atributa i operativnih informacija, kako bi sadržali svojstva i mogućnosti stvarnog sveta. Važna je karakteristika da u objektno-orijentisanoj bazi podataka rečnik podataka ne samo da skladišti relacije među objektima, već i razume ponašanje objekta.

Ukoliko su objekti slični jedan drugom u ponašanju i nekim drugim atributima, mogu se zajedno svrstati u **klasu**. Klasu treba shvatiti kao šablon za objekte, što može pomoći u radu sa veoma složenim objektima. Mogućnost grupisanja po klasama omogućava da se iskoriste sličnosti u ponašanju među objektima druge karakteristike, koje su objektima zajedničke.

Jedna od osnovnih stvari vezanih za objektno-orijentisani model je podrška učenja. Učenje nastaje kada se podaci vezuju za određeni objekat, tako da im se može pristupiti samo ako objekat odobri, ili prihvati. Direktna korist od ovoga je zaštićenost podataka od nelegalnog pristupa.

Projektanti baze podataka smatraju da objekti predstavljaju učenje podataka i ponašanja, kao i to da objekti predstavljaju skup kôdova i podataka koji mogu da funkcionišu nezavisno, što je veoma moćno svojstvo. Ono što je takođe bitno je da se smatra da relaciono pravilo o nezavisnosti podataka ne narušava učenje.

Učtaurenje uključuje mogućnost pristupa objektima i njihovim podacima samo kroz postupke koje je objekat prihvatio. Ovo je u suprotnosti sa osnovnim principom relacionog modela baze podataka, poznatijim kao nezavisnost podataka (*prema relacionom modelu koji su definisali Codd i Date, podacima se može pristupiti nezavisno i onda kada se to želi*).

Čini se da ove razlike čine relacioni model nekompatibilnim sa objektno-orijentisanim modelom, ali ipak to nije slučaj. U objektno-orijentisanom modelu ponašanje podataka čuva se u bazi podataka i ne nalazi se izvan nje. Pošto se ponašanje podataka čuva u bazi podataka, ova šema ne dovodi u opasnost nezavisnost aplikacija od podataka. Ovo je osnova definicije o nezavisnosti podataka.

Mogućnost **proširenja** je sposobnost objektno-orijentisanog modela baze podataka da dodaje nove objekte i njima svojstvene postupke, bez ikakvog uticaja na ostale objekte i aplikacije. Pošto podaci mogu da se učeure sa objektima, ova sposobnost omogućava objektnom modelu da rešava nestandardne situacije u kojima podaci mogu da se nađu.

Nasleđivanje se može shvatiti kao način podele kôda. Nova klasa objekata se može definisati u granicama postojeće, odnosno osnovne klase. Kako se kreiraju objekti nižeg reda, oni nasleđuju ili pristupaju podacima i postupcima koji su u vezi sa svim klasama iznad njih. Klasa objekata koja se izvodi iz osnovne klase obično proširuje ili redefiniše postojeću strukturu i ponašanje osnovne klase.

Drugi način razumevanja nasleđivanja je da nove klase nastaju kao tipične specijalizacije postojećih klasa. Može se pomenuti i to da postoje dva različita tipa nasleđivanja i to:

- ✓ *nasleđivanje podataka* – kada deo podataka može da nasledi dodatne attribute ostalih podataka iz klase;
- ✓ *nasleđivanje funkcije ili objekta* – kada objekat može da nasledi podatke i attribute od drugog objekta iz klase.

Polimorfizam je sposobnost dva različita objekta da se ponašaju različito kada dobiju istu poruku. Koncept polimorfizma postoji u stvarnom svetu, u kome se istovetne poruke dobijaju svakog dana, a na njih se ipak različito reaguje. Dakle, objekat prema informacijama koje dobija reaguje različito i razume kontekst informacija koji dobija kao input.