

UNIVERZA V LJUBLJANI  
EKONOMSKA FAKULTETA

ZAKLJUČNA STROKOVNA NALOGA VISOKE POSLOVNE ŠOLE  
**PRIMERJAVA SISTEMOV ZA UPRAVLJANJE S PODATKOVNIMI  
BAZAMI**

Ljubljana, avgust 2022

JANJA BREZOVNIK

## IZJAVA O AVTORSTVU

Podpisana Janja Brezovnik, študentka Ekonomske fakultete Univerze v Ljubljani, avtorica predloženega dela z naslovom Primerjava sistemov za upravljanje s podatkovnimi bazami, pripravljene v sodelovanju s svetovalcem doc. dr. Luko Tomatom

### IZJAVLJAM

1. da sem predloženo delo pripravila samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbela, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobila vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označila;
7. da sem pri pripravi predloženega dela ravnala v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobila soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve predloženega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi.

V Ljubljani, dne \_\_\_\_\_

Podpis študentke: \_\_\_\_\_

# KAZALO

<b>UVOD</b> .....	<b>1</b>
<b>1</b> <b>PODATKOVNE BAZE</b> .....	<b>2</b>
<b>1.1</b> <b>Opredelitev podatkovnih baz</b> .....	<b>2</b>
<b>1.2</b> <b>Razvoj podatkovnih baz</b> .....	<b>2</b>
<b>1.3</b> <b>Vrste podatkovnih baz</b> .....	<b>5</b>
1.3.1 Relacijski podatkovni model .....	5
1.3.2 Hierarhični podatkovni model .....	5
1.3.3 Mrežni podatkovni model.....	6
1.3.4 Objektni podatkovni model .....	6
1.3.5 Večdimenzionalni podatkovni model.....	7
1.3.6 Stolpčno usmerjeni podatkovni model .....	7
1.3.7 NoSQL podatkovna baza.....	7
<b>1.4</b> <b>Področja uporabe podatkovnih baz</b> .....	<b>8</b>
<b>2</b> <b>SISTEMI ZA UPRAVLJANJE S PODATKOVNIMI BAZAMI</b> .....	<b>8</b>
<b>2.1</b> <b>Predstavitev sistemov za upravljanje s podatkovnimi bazami</b> .....	<b>8</b>
<b>2.2</b> <b>Razvoj sistemov za upravljanje s podatkovnimi bazami</b> .....	<b>9</b>
<b>2.3</b> <b>Naloge sistemov za upravljanje s podatkovnimi bazami</b> .....	<b>10</b>
<b>2.4</b> <b>Prednosti upravljanja podatkovnih baz</b> .....	<b>10</b>
<b>3</b> <b>PRIMERJAVA SISTEMOV ZA UPRAVLJANJE S PODATKOVNIMI BAZAMI</b> .....	<b>11</b>
<b>3.1</b> <b>MySQL</b> .....	<b>11</b>
<b>3.2</b> <b>Oracle Database</b> .....	<b>11</b>
<b>3.3</b> <b>Microsoft SQL Server</b> .....	<b>12</b>
<b>3.4</b> <b>IBM DB2</b> .....	<b>12</b>
<b>3.5</b> <b>Microsoft Access</b> .....	<b>13</b>
<b>3.6</b> <b>PostgreSQL</b> .....	<b>13</b>
<b>3.7</b> <b>Primerjava DBMS</b> .....	<b>13</b>
<b>4</b> <b>DISKUSIJA</b> .....	<b>15</b>
<b>4.1</b> <b>Smernice za nadaljnje delo</b> .....	<b>17</b>
<b>SKLEP</b> .....	<b>17</b>
<b>LITERATURA IN VIRI</b> .....	<b>18</b>

## KAZALO TABEL

Tabela 1: Osnovne informacije DBMS .....	14
Tabela 2: Podprti operacijski sistemi DBMS .....	14
Tabela 3: Temeljne značilnosti DBMS .....	14
Tabela 4: Omejitve DBMS.....	15

## SEZNAM KRATIC

angl. – angleško

**ACID** – (angl. atomicity, consistency, isolation, durability); atomičnost, doslednost, izolacija in trajnost

**ANSI** – (angl. American national standards institute); Ameriški državni inštitut za standarde

**API** – (angl. Application programming interface); aplikacijski programski vmesnik

**CODASYL** – (angl. Conference on data systems language); konferenca za jezike podatkovnih sistemov

**DBMS** – (angl. Database management system); sistem za upravljanje podatkovnih baz

**DBTG** – (angl. Data Base task group); delovna skupina Data Base

**GDPR** – (angl. General data protection regulation); splošna uredba o varstvu podatkov

**GUAM** – (angl. Generalized update access method), splošna metoda dostopa do posodobitev

**GUI** – (angl. Graphical user interface); grafični uporabniški vmesnik

**I/O** – (angl. input/output); vhod/izhod

**IDS** – (angl. Integrated data store); integrirana podatkovna baza

**IMS** – (angl. Information management system); sistem za upravljanje informacij

**INGRES** – (angl. Interactive graphics and retrieval system); interaktiven grafični sistem za upravljanje zbirk podatkov

**ISO** – (angl. International organization for standardization); Mednarodna organizacija za standardizacijo

**NIS** – (angl. Network and information security); direktiva o varnosti podatkov

**ODBC** – (angl. Open database connectivity); metoda za odprto podatkovno povezljivost

**OLAP** – (angl. Online analytical processing); spletna analitična obdelava

**OLTP** – (angl. Online transaction processing); spletna obdelava transakcij

**SQL** – (angl. Structured query language); strukturirani jezik poizvedb

## UVOD

Podatki že od nekdaj igrajo pomembno vlogo na vseh področjih človekovega razvoja, naj gre za šport, zdravje, izobraževanje, kulturo itd. Omogočili so analiziranje in oblikovanje rešitev ali preprosto razvijanje načrtov, ki so danes po zaslugi tehnologije shranjeni v organiziranih zbirkah, imenovanih podatkovne baze. Z razvojem procesov so se hitro razvijale metode shranjevanja, analize in vzdrževanja podatkov. Zaradi tega je podatkovna baza postala zelo pomemben vidik v vseh procesih, pravilno razumevanje in vzdrževanje podatkov pa sta postala bistvena za razvoj poslovanja in produktov (Chauhan, Kumar & Kohli, 2017).

Elmasri in Navathe (2007) sta dejala, da se tekom dneva srečujemo z več dejavnostmi, ki vključujejo nezavedno interakcijo s podatkovno bazo. Ko na banko položimo ali z nje dvignemo sredstva, opravimo rezervacijo letalskega prevoznika, dostopamo do računalniško podprtega knjižničnega sistema, kupimo kakšen predmet npr. knjigo, igračo ali računalnik, je velika verjetnost, da bodo naše dejavnosti vključevale računalniški program, ki z uporabo sistema za upravljanje s podatkovnimi bazami (angl. Database management system, v nadaljevanju DBMS), dostopa do baze podatkov. V zadnjih nekaj letih je napredek v tehnologiji vodil do novih aplikacij in sistemov baz podatkov. Večpredstavnostne baze podatkov lahko zdaj shranjujejo slike, video posnetke in zvočna sporočila. Geografski informacijski sistemi lahko shranjujejo in analizirajo zemljevide, vremenske podatke in satelitske slike. Podatkovna skladišča (angl. Data warehouse) in sistemi za spletno analitično obdelavo (angl. Online analytical processing, v nadaljevanju OLAP) se v mnogih podjetjih uporabljajo za pridobivanje in analizo koristnih informacij iz zelo velikih podatkovnih baz za sprejemanje odločitev. Tehnologija aktivne baze podatkov se v realnem času uporablja pri nadzoru industrijskih in proizvodnih procesov. Na svetovnem spletu se uporabljajo tehnike iskanja po bazi, za izboljšanje iskanja informacij, ki jih potrebujejo uporabniki, ki brskajo po internetu.

Namen zaključne naloge je opredeliti in predstaviti podatkovne baze in izbrane DBMS ter jih med sabo primerjati. Ugotoviti želim tudi, na kakšen način koristijo podjetjem.

Zastavljeni cilji v teoretičnem delu so pregled literature, ki je relevantna za ustrezne definicije in predstavbo DBMS. Literaturo sem smiselno povezala in predstavila prednosti, slabosti, podobnosti ter razlike med posameznimi sistemi.

Zaključna strokovna naloga je sestavljena iz teoretičnega in empiričnega dela. V teoretičnem delu sem podrobneje predstavila podatkovne baze, vrste podatkovnih baz, DBMS in njihovo uporabo. V empiričnem delu pa sem s primerjalno metodo primerjala izbrane DBMS z vidika osnovnih informacij, integracij z aplikacijami, značilnosti in zmogljivosti oziroma omejitev. V prvem poglavju sem se osredotočila na podatkovno bazo, razvoj podatkovnih baz in njihove vrste. V drugem poglavju nadaljujem z razvojem DBMS, njihovo uporabo in prednostmi. Tretje poglavje členim na opis šestih DBMS in

njihovo primerjavo. Četrto poglavje je namenjeno ključnim ugotovitvam, pomenu DBMS za podjetja, omejitvam raziskave in smernicam za nadaljnje delo. Vsebino zaključne strokovne naloge sem na ustreznih mestih podprla s tabelami, kjer je to potrebno.

## **1      PODATKOVNE BAZE**

V tem poglavju se bom osredotočila na podrobno opredelitev podatkovnih baz. Predstavila bom njihovo opredelitev, uporabo, razvoj in vrste.

Za razumevanje definicije podatkovnih baz je najprej potrebno razumeti razliko med izrazoma »podatki« in »informacije«. Podatki so definirani kot posamezna dejstva, informacije pa so organizacija in interpretacija teh dejstev, ki podatke postavljajo v kontekst (Hill, 2021). Podatki so surovi in neorganizirani, informacije pa organizirane.

### **1.1    Opredelitev podatkovnih baz**

Podatkovne baze so organizirana zbirka strukturiranih informacij ali podatkov, ki so po navadi elektronsko shranjeni v računalniški sistem. Bazo podatkov običajno nadzira DBMS. Podatki in DBMS se skupaj s povezanimi aplikacijami, običajno krajše imenujejo podatkovni sistem ali baza podatkov (Oracle, brez datuma a). Baza podatkov se nanaša na povezane podatke v strukturirani obliki. Za učinkovitejšo obdelavo in poizvedovanje, so podatki v podatkovnih bazah organizirani v tabele, sestavljene iz vrstic in stolpcev. Indeksirani so tako, da je mogoče do njih enostavno dostopati, jih upravljati, posodabljati, nadzorovati, organizirati ali izbrisati (Geeks for Geeks, 2021).

»Podatek« definiramo kot vrednost atributa entitete. Vsako zbirko povezanih podatkovnih postavk subjektov z enakimi atributi imenujemo »baza podatkov«. Zgolj zbiranje podatkov še ne pomeni podatkovne baze. O bazi podatkov lahko govorimo takrat, ko so podatki v njej organizirani na način, ki omogoča njihovo učinkovito uporabo (Gunjal, 2003). Podjetjem omogoča, da vse bistvene informacije upravljajo na enem mestu. Prihrani ogromno časa pri iskanju pomembnih informacij o njihovem poslovnem napredku. Podatkovne baze bistveno olajšajo tudi upravljanje s človeškimi viri (European Business Review, 2020).

### **1.2    Razvoj podatkovnih baz**

Računalniške podatkovne baze so se začele v šestdesetih letih prejšnjega stoletja, ko je uporaba računalnikov postala stroškovno učinkovitejša možnost za zasebne organizacije. Zgodovina računalnikov in podatkovnih baz sta med seboj povezani. S padcem cen strojne opreme je postalo lažje prenesti shranjevanje podatkov in podatkovnih baz v računalnike (Quickbase, brez datuma).

Prvo računalniško bazo podatkov, znano kot integrirana podatkovna baza (angl. Integrated data store – IDS), je zasnoval Charles Bachman. Kmalu ji je sledil sistem za upravljanje informacij (angl. Information management system – IMS), baza podatkov, ki jo je ustvaril IBM. Obe bazi podatkov sta bili predhodnici navigacijske baze podatkov. Navigacijske baze podatkov so od uporabnikov zahtevale krmarjenje po celotni bazi, da bi našli željene informacije. Pri tem sta obstajala dva glavna modela: hierarhični in mrežni model. Hierarhični model je razvil IBM. V njem so podatki organizirani kot družinsko drevo. Vsak vnos podatkov ima nadrejeni zapis, ki se začne z korenskim zapisom. Mrežni model je bil medtem objavljen na konferenci za jezike podatkovnih sistemov (angl. Conference on data systems languages, v nadaljevanju CODASYL). Od hierarhičnega modela se je razlikoval po tem, da je zapisu omogočal več kot en nadrejen in en podrejen zapis (Think Automation, brez datuma).

V sedemdesetih letih prejšnjega stoletja je Codd (v Quickbase, brez datuma) objavil pomemben članek, ki je predstavljal osnovo za razvoj relacijske baze podatkov. Njegove ideje so spremenile način razmišljanja o podatkovnih bazah. Shema baze podatkov ali logična organizacija je v njegovem modelu ločena od fizičnega shranjevanja informacij, kar je postalo standardno načelo za podatkovne sisteme.

Relacijska baza podatkov prikazuje razmerje med različnimi zapisi podatkov. Za razliko od njenih navigacijskih dvojnikov, bi bilo po relacijskih bazah podatkov mogoče iskati, prav tako bi bile bolj prostorsko učinkovite, kar zmanjša stroške shranjevanja podatkov. Sledilo je ustvarjanje interaktivnega grafičnega sistema za upravljanje zbirk podatkov (angl. Interactive graphics and retrieval system, v nadaljevanju INGRES) na kalifornijski univerzi Berkeley. INGRES je bil model relacijske baze podatkov, ki je dokazal izvedljivost Coddovih idej. Uporabljal je poizvedovalni jezik imenovan QUEL. Kasneje je IBM izdal svoj pogled na relacijsko bazo podatkov, poznan kot System R. System R je bil prvi v zgodovini baz podatkov, ki je uporabljal strukturirani jezik poizvedb (angl. Structured query language, v nadaljevanju SQL) (Think Automation, brez datuma). SQL je postal standardni jezik poizvedb, ki sta ga izbrala Ameriški državni inštitut za standarde (angl. American national standards institute, v nadaljevanju ANSI) leta 1986 in Mednarodna organizacija za standardizacijo (angl. International organization for standardization, v nadaljevanju ISO) leta 1987 ter je v uporabi še danes. Sistemi relacijskih baz podatkov so postali velik uspeh, saj je hitro povečanje prodaje računalnikov spodbudilo trg podatkovnih baz in posledično velik upad priljubljenosti mrežnih in hierarhičnih modelov baz podatkov. DB2 je postal vodilni izdelek baze podatkov za IBM, uvedba IBM PC-ja pa je povzročila ustanovitev številnih novih podjetij za baze podatkov in razvoj izdelkov (Quickbase, brez datuma).

Drugi pomemben dogodek v zgodovini baz podatkov je bil pojav objektno usmerjenih sistemov za njihovo upravljanje. Ta koncept se je pojavil v sredini osemdesetih let prejšnjega stoletja. Objektne baze podatkov obravnavajo podatek kot »predmet« in

uporabljajo programske jezike, ki podpirajo objektno-usmerjen pristop (Think Automation, brez datuma).

Devetdeseta leta so imela ključno vlogo pri napredku in programski opremljenosti baz podatkov. Podobno kot v šestdesetih letih prejšnjega stoletja je širši kulturni premik vodil k nadaljnemu razvoju industrije. V tem času so bila izdana nova odjemalska orodja za razvoj aplikacij, ki vključujejo Oracle Developer, PowerBuilder, VB in druge. Razvita so bila tudi številna orodja za osebno produktivnost, kot sta metoda za odprto podatkovno povezljivost (angl. Open database connectivity – ODBC) in MS Access (Quickbase, brez datuma).

Sredi devetdesetih let prejšnjega stoletja je pojav interneta povzročil eksponentno rast industrije baz podatkov. Povprečni uporabniki namiznih računalnikov so začeli uporabljati sisteme baz podatkov odjemalec-strežnik za dostop do računalniških sistemov, ki so vsebovali stare podatke. Ker je vedno več uporabnikov kupovalo osebne računalnike in se posluževalo spleta, je rastle potreba po izboljšanju podatkovnih baz (Quickbase, brez datuma).

Proti koncu devetdesetih let prejšnjega stoletja je povečano vlaganje v spletna podjetja povzročilo povečanje povpraševanja po priključkih za internetne baze podatkov, kot so Front Page, Active Server Pages, Java Servlets, Dream Weaver, ColdFusion, Enterprise Java Beans in Oracle Developer 2000. Uporaba CGI, GCC, MySQL, Apache in drugih sistemov je preko interneta omogočala uporabo odprtokodnih rešitev (Quickbase, brez datuma).

Odprtokodna zbirka podatkov je zbirka podatkov, kjer si lahko kdorkoli preprosto ogleda izvorno kodo. Je odprta in jo je mogoče brezplačno prenesti. Tudi za različico skupnosti so naloženi nekateri majhni dodatni in dostopni stroški. Odprtokodna zbirka podatkov zagotavlja omejeno tehnično podporo končnim uporabnikom. Namestitve in posodobitve upravlja uporabnik. Komerzialne ali zaprtokodne zbirke podatkov (lastniške) so bile ustvarjene samo za komercialne namene, so »premium« in niso brezplačne kot odprtokodne zbirke podatkov. V zaprtokodnih zbirkah je zagotovljena tehnična podpora. Namestitve in posodobitve upravlja prodajalec programske opreme (Geeks for Geeks, brez datuma).

Zaradi povečane uporabe tehnologije na prodajnem mestu, sta bila vse bolj zaželeni spletna obdelava transakcij (angl. Online transaction processing – OLTP) in OLAP (Quickbase, brez datuma).

Leta 1988 je Carlo Strozzi prvič uporabil izraz NoSQL, ko je poimenoval svojo lahko, odprtokodno »relacijsko« bazo podatkov, ki ni uporabljala SQL. Ime se je kasneje ponovno pojavilo leta 2009, ko sta ga Eric Evans in Johan Oskarsson uporabila za opis nerelacijskih baz podatkov (Foote, 2018). Baze podatkov NoSQL so uporabne za



nestrukturirane podatke in so v dvatisočih letih zabeležile rast. Omogočile so hitrejšo obdelavo večjega, bolj raznolikega nabora podatkov.

Od leta 2010 dalje je prišlo do porasta masovnih podatkov (angl. Big data), povečane ozaveščenosti o podatkih in večjega poudarka na njihovem varstvu. Masovni podatki posledično potrebujejo velike baze podatkov za njihovo shranjevanje. Zaradi potrebe po zbiranju, organiziranju in uporabi tako velikih količin podatkov, je programska oprema za avtomatizacijo postala priljubljeno orodje pri interakciji z bazami podatkov. Splošna uredba o varstvu podatkov (angl. General data protection regulation – GDPR) in direktiva o varnosti omrežij in formacij (angl. Network and information security – NIS) sta služili kot dodatno poudarjanje pomena varovanja podatkov. Veliko pozornosti je bilo namenjene porazdeljenim bazam podatkov zaradi globalizacije. Podatke so hranili na več fizičnih lokacijah in ne samo na enem mestu (Think Automation, brez datuma).

### **1.3 Vrste podatkovnih baz**

#### 1.3.1 Relacijski podatkovni model

Relacijska baza podatkov je vrsta baze podatkov, ki shranjuje in omogoča dostop do medsebojno povezanih podatkovnih točk. Relacijske baze podatkov temeljijo na relacijskem modelu, intuitivnem, enostavnem načinu predstavljanja podatkov v tabelah. V relacijski bazi podatkov je vsaka vrstica v tabeli zapis z edinstveno vrednostjo oziroma ID-jem, imenovanim ključ. Stolpci tabele vsebujejo attribute podatkov in vsak zapis ima običajno vrednost za vsak atribut, kar olajša vzpostavitev razmerij med podatkovnimi točkami (Oracle, brez datuma b). Stolpci za tabelo strank so lahko ID stranke, ime podjetja, naslov podjetja itd., stolpci za tabelo transakcij pa datum transakcije, znesek transakcije, ID stranke, način plačila itd. V tem primeru so tabele med seboj povezane na podlagi skupnega polja ID stranke, kar uporabnikom omogoča poizvedovanje po tabeli in ustvarjanje poročil (IBM Cloud Education, 2019).

Z relacijskimi bazami podatkov komuniciramo z uporabo jezika SQL. V razvoju je od leta 1986 in, čeprav je določen s standardom ANSI/ISO, danes poznamo več različic (Kozubek-Krycuń, 2022).

#### 1.3.2 Hierarhični podatkovni model

Chauhan, Kumar in Kohli (2017) so hierarhični model opredelili kot drevesno strukturo, ki služi istemu namenu kot entiteta – razmerje (angl. entity – relationship), v kateri so zapisi shranjeni v skupinah »starš« (angl. Parent) in »otrok« (angl. Child) ter med njima vzdržujejo predvsem razmerja »ena proti mnogo«. Ta model zbira podatke v obliki zapisov, za katere je mogoče navesti, da so enakovredni tabelam, pri čemer so posamezni zapisi enakovredni vrsticam.

Po zgradbi je podoben računalniškemu datotečnemu sistemu (Slv.kagutech.com, 2022). Razmerje »ena proti mnogo« pomeni, da ima lahko nadrejeni zapis več podrejenih zapisov, podrejeni zapis pa lahko ima samo en nadrejeni zapis. Zapisi so povezani preko povezav. Vrsta zapisa pove, katero polje ga vsebuje. Vsako polje lahko vsebuje samo eno vrednost. Pridobivanje podatkov iz hierarhičnega podatkovnega modela zahteva pregled celotnega »drevesa« z začetkom v korenskem zapisu (Heavy.ai, brez datuma).

### 1.3.3 Mrežni podatkovni model

Mrežni podatkovni model omogoča, da se več zapisov poveže z isto datoteko lastnika. Za lažjo predstavo si ga je mogoče predstavljati kot narobe obrnjeno drevo, kjer so veje, ki so informacije o članu, povezane z lastnikom, ki je dno drevesa. Mrežna podatkovna baza je zelo prilagodljiva. Razmerje, ki ga imajo informacije v mrežnem podatkovnem modelu, je opredeljeno kot »mного proti mnogo«. To pomeni, da je ena datoteka lastnika lahko povezana z mnogimi datotekami članov ali obratno (Computer Business Research, brez datuma).

Razvit je bil kot izboljšava že obstoječega hierarhičnega podatkovnega modela. Ker je bil model hierarhične baze podatkov zelo pomanjkljiv, se je Charles Bachman odločil ustvariti bazo podatkov, ki je podobna hierarhični bazi podatkov, vendar z večjo prilagodljivostjo in manj privzetimi nastavitvami. Izvirna in obstoječa hierarhična baza podatkov ima eno datoteko lastnika, ki je strogo povezana z eno datoteko člana, kar ustvarja učinek lestve, ki je omejeval zbirko podatkov pri iskanju odnosov zunaj svoje kategorije (Computer Business Research, brez datuma).

### 1.3.4 Objektni podatkovni model

Objektno usmerjene podatkovne baze so se pojavile, da bi zadovoljile potrebo po povezovanju objektno usmerjenih programskih jezikov z bazo podatkov. Čeprav objektno usmerjene baze podatkov obstajajo že od poznih sedemdesetih let prejšnjega stoletja, so bile v zadnjih desetletjih razmeroma slabo sprejete zaradi vse večje razširjenosti funkcionalnih programskih jezikov in relacijskih baz podatkov. V zadnjem času se njihova uporaba povečuje, predvsem zaradi možnosti hitrih poizvedb in enostavne kode (MongoDB, Inc., brez datuma).

Objektne baze podatkov so tesno povezane z določenim objektno usmerjenim programskim jezikom. Omogočajo trajno shranjevanje podatkov v objektih, brez potrebne pretvorbe v relacijsko bazo podatkov (MongoDB, Inc., brez datuma). Entiteta je predstavljena kot objekt, predmeti so shranjeni v pomnilniku. Objekti imajo člane, kot so polja, lastnosti in metode. Imajo tudi življenjski cikel, ki vključuje ustvarjanje predmeta, njegovo uporabo in izbris. Danes obstaja veliko priljubljenih jezikov, ki so objektno usmerjeni. Mednje sodijo C++, Java, C#, Ruby, Python, JavaScript in Perl (C# Corner,

2019). Aplikacija lahko deluje z manj kode, uporablja bolj naravno modeliranje podatkov in ustvarja baze kod, ki jih je lažje vzdrževati. Izpeljava objektno usmerjene baze podatkov je celovitost sistemov objektno usmerjenih programskih jezikov in konsistentnih sistemov. Ta model je zelo obsežen v obsegu obdelave podatkov in kode, vendar so bili stroški razvoja in vzdrževanja zelo visoki v primerjavi z relacijskimi DBMS, kar je omejilo uporabo te metode (Chauhan, Kumar & Kohli, 2017).

### 1.3.5 Večdimenzionalni podatkovni model

Večdimenzionalno podatkovno bazo lahko kategoriziramo za shranjevanje podatkov in OLAP. Sestavljena je iz več relacijskih baz podatkov. Večdimenzionalni podatki se uporabljajo za analize in poročila, ki služijo kot primarna naloga za procese poslovnega odločanja (Pedamkar, brez datuma). Večdimenzionalna podatkovna baza omogoča hitro obdelavo podatkov in posledično tudi hitro ustvarjanje odgovorov. Uporabnikom omogoča postavljanje analitičnih vprašanj o poslovanju in tržnih trendih, vse za sprejemanje učinkovitih odločitev o strateških usmeritvah organizacije (Gillis, brez datuma).

### 1.3.6 Stolpčno usmerjeni podatkovni model

Stolpčno usmerjeni podatkovni model podatkov se ne hrani v vrsticah, temveč v stolpcih. Je zbirka podatkov, ki shranjuje vrednosti vsakega stolpca skupaj, namesto shranjevanja vrednosti vsake vrstice skupaj (ScyllaDB, brez datuma). Namen stolpčno usmerjene podatkovne baze je učinkovito pisanje podatkov na in branje podatkov iz pomnilnika trdega diska, da se skrajša čas, potreben za vrnitev poizvedbe. Stolpčne podatkovne baze hranijo podatke na način, ki močno izboljša zmogljivost vhod/izhod (angl. input/output – I/O) diska (Tech Target Contributor, 2021). Podatki, do katerih uporabniki dostopajo ločeno, se lahko hranijo v različnih družinah stolpcev. Znotraj družin je mogoče dinamično dodajati nove stolpce. Ni nujno, da mora imeti vrstica vrednost za vsak stolpec (Kumar, 2021).

Ena izmed glavnih prednosti stolpčno usmerjene podatkovne baze, v primerjavi z vrstično usmerjeno, je hitrejša zmogljivost, zaradi dostopa do manj pomnilnika za izhodne podatke. Stolpčna baza podatkov shranjuje podatke po stolpcih namesto po vrsticah, zato lahko shrani več podatkov v manjši količini pomnilnika. Začetno pridobivanje podatkov se izvaja na podlagi stolpca za stolpcem, zato se pridobijo samo stolpci, ki jih je potrebno uporabiti. To omogoča učinkovito povečanje stolpčne baze podatkov in obdelovanje velike količine podatkov (Tech Target Contributor, 2021).

### 1.3.7 NoSQL podatkovna baza

NoSQL podatkovne baze so namensko zgrajene za posebne podatkovne modele in imajo prilagodljive sheme za gradnjo sodobnih aplikacij. Priznane so predvsem zaradi njihovega

enostavnega razvoja, funkcionalnosti in zmogljivosti v velikem obsegu (Amazon Web Services, Inc., brez datuma). Baze podatkov NoSQL se imenujejo tudi »nerelacijske« oziroma »ne-SQL« baze podatkov. Z imenom se poudari dejstvo, da lahko obravnavajo velike količine hitro spreminjajočih se nestrukturiranih podatkov na različne načine kot relacijske (SQL) podatkovne baze (Microsoft, brez datuma).

Baze podatkov NoSQL so bolj prilagodljive kot tradicionalne relacijske baze podatkov (Think Automation, brez datuma), kar je vodilo organizacije, kot so Facebook, Twitter, LinkedIn in Google, da so pričele uporabljati sisteme NoSQL. Te organizacije obdelujejo ogromne količine nestrukturiranih podatkov in jih usklajujejo, da najdejo vzorce in pridobijo poslovne vpoglede. Masovni podatki je postal uradni izraz leta 2005 (Foote, 2018).

#### **1.4 Področja uporabe podatkovnih baz**

Podatkovne baze so močna orodja, ki se uporabljajo na vseh področjih računalništva. So ključna računalniška veščina za doseg organiziranja podatkov, ustvarjanja baz podatkov in njihovega nadzora z uporabo poizvedovalnih jezikov (BBC, brez datuma).

Upravljanje informacije pomeni skrbeti zanje tako, da delujejo za uporabnike in so uporabne za naloge, ki jih opravljajo. Z uporabo DBMS podatki, ki jih uporabniki zbirajo in dodajajo v njegovo bazo podatkov, niso več del naključne neorganiziranosti (Watt, brez datuma).

Baze podatkov se uporabljajo skoraj povsod, vključno z bankami, maloprodajo, spletnimi mesti in skladišči. Banke jih uporabljajo za spremljanje računov strank, stanja in depozitov. Maloprodajne trgovine lahko uporabljajo zbirke podatkov za shranjevanje cen, informacij o strankah, prodajnih informacij in količin. Spletna mesta jih uporabljajo za shranjevanje vsebine, podatkov za prijavo strank in njihovih preferenc. Skladišča uporabljajo baze podatkov za upravljanje ravni zalog in lokacije skladiščenja. Baze podatkov se uporabljajo povsod, kjer je treba podatke shraniti in jih enostavno pridobiti (Graham, brez datuma).

## **2 SISTEMI ZA UPRAVLJANJE S PODATKOVNIMI BAZAMI**

### **2.1 Predstavitev sistemov za upravljanje s podatkovnimi bazami**

Sumathi in Esakkirajan (2007) sta opredelila DBMS kot zbirko medsebojno povezanih podatkov in niz programov za njihov dostop. Gre za programsko opremo, ki služi za pomoč pri vzdrževanju in uporabi baze podatkov. DBMS je sestavljen iz zbirke medsebojno povezanih in trajnih podatkov, ki se imenuje baza podatkov, in nabora aplikacijskih programov, ki se uporabljajo za dostop, posodabljanje in upravljanje

podatkov. Ta del predstavlja sistem za upravljanje podatkov (angl. Management system – MS).

DBMS je programska oprema splošnega namena, kar pomeni, da ni specifična za aplikacijo. Isti sistem (npr. Oracle ali MySQL) se lahko uporablja v hotelskem rezervacijskem sistemu, upravljanju hotela ali knjižnice. Skrbi za shranjevanje in dostop do podatkov, aplikacijskim programom pa prepušča le naloge, specifične za aplikacijo (Sumathi & Esakkirajan, 2007).

DBMS je zapleten sistem, ki uporabniku omogoča vnašanje, skupno rabo, urejanje, manipulacijo podatkov in njihov prikaz v podatkovni bazi. Kompleksnost DBMS sega do njegovega oblikovanja in izvedbe, saj večjemu številu uporabnikov omogoča delitev podatkov (Sumathi & Esakkirajan, 2007).

## **2.2 Razvoj sistemov za upravljanje s podatkovnimi bazami**

Po Sumathi in Esakkirajan (2007) je bil predhodnik DBMS sistem, ki temelji na datotekah. Proces pristanka Apolla na Luni se je začel leta 1960, ko še ni bilo na voljo sistema za obdelavo in upravljanje velikih količin informacij. Posledično je ameriško letalsko-vesoljsko podjetje North American Aviation, danes znano kot Rockwell International, razvilo programsko opremo splošne metode dostopa do posodobitev (angl. Generalized update access method, v nadaljevanju GUAM).

Sredi šestdesetih let prejšnjega stoletja se je IBM pridružil podjetju North American Aviation, da bi razvil GUAM v informacijski sistem za upravljanje. Ta je temeljil na hierarhičnem podatkovnem modelu. Leta 1967 je CODASYL ustanovila delovno skupino Data Base (angl. Data Base task group – DBTG), ki je določila tri različne jezike za standardizacijo. To so jezik za definiranje podatkov (angl. Data definition language, v nadaljevanju DDL), ki bi skrbniku baze podatkov omogočil definiranje sheme, podshemo DDL, ki bi aplikacijskim programom omogočila definiranje delov baze podatkov in jezik za manipuliranje podatkov (angl. Data manipulation language – DML) (Sumathi & Esakkirajan, 2007).

V osemdesetih letih prejšnjega stoletja je IBM izdal dva komercialna sistema za upravljanje relacijskih baz podatkov, znana kot DB2 in SQL/DS, Oracle Corporation pa je izdal Oracle. Leta 1979 je Edvard Codd sam poskušal odpraviti nekatere pomanjkljivosti v svojem izvirnem delu z razširjeno različico relacijskega modela, imenovanega RM/T leta 1979 in RM/V2 leta 1990 (Sumathi & Esakkirajan, 2007). Poskusi zagotavljanja podatkovnega modela, ki predstavlja »resnični svet«, so bili definirani kot modeliranje semantičnih podatkov. Semantični podatkovni model je metoda strukturiranja podatkov, da se predstavijo na specifičen logični način. Je konceptualni podatkovni model, ki vključuje semantične informacije, ki dodajajo osnovni pomen podatkom in razmerjem, ki ležijo med njimi (Techopedia, 2020).

## 2.3 Naloge sistemov za upravljanje s podatkovnimi bazami

Od DBMS naj bi bilo po Garcia-Molina, Ullman in Widom (2008) pričakovano, da omogoča uporabnikom ustvarjanje nove baze podatkov in določanje njihove sheme ter logične strukture z uporabo specializiranega jezika. Poleg tega naj bi uporabnikom omogočal možnost poizvedovanja po podatkih s pomočjo poizvedb, kot tudi spremembo podatkov z uporabo ustreznega poizvedovalnega jezika oziroma jezika za obdelavo podatkov. DBMS mora podpirati shranjevanje velikih količin podatkov skozi daljše časovno obdobje z namenom učinkovitega dostopa do podatkov in sprememb podatkovne baze. Omogočati mora možnost obnovitve baze podatkov ob morebitnih napakah ali namerni zlorabi ter nadzirati dostop do podatkov s strani več uporabnikov hkrati.

## 2.4 Prednosti upravljanja podatkovnih baz

Ramakrishnan in Gehrke (2011) sta kot prednosti uporabe DBMS opredelila:

- **Neodvisnost podatkov** (angl. Data independence) – aplikacijski programi bi morali biti čim bolj neodvisni od podrobnosti o predstavitvi in shranjevanju podatkov. DBMS lahko zagotovi abstrakten pogled na podatke, tako da izolira kodo aplikacije od takšnih podrobnosti. Podatki so ločeni od programov, zato spremembe podatkov ne vplivajo na izvajanje programa in aplikacijo.
- **Učinkovit dostop do podatkov** (angl. Efficient data access) – DBMS uporablja različne tehnike za učinkovito shranjevanje in pridobivanje podatkov. Ta funkcija je še posebej pomembna, če so podatki shranjeni na zunanjih napravah za shranjevanje.
- **Celovitost in varnost podatkov** (angl. Data integrity and security) – varnost podatkov se nanaša na zaščito podatkov pred nepooblaščenim dostopom ali poškodovanjem in je potrebna za zagotavljanje celovitosti podatkov. Če se do podatkov vedno dostopa preko DBMS, lahko DBMS uveljavi omejitve integritete podatkov. Denimo, preden uporabnik v sistem vnese plačo zaposlenega, lahko DBMS preveri, da proračun oddelka še ni bil presežen. Prav tako lahko DBMS uveljavi nadzor dostopa, ki ureja, kateri podatki so vidni različnim razredom uporabnikov.
- **Upravljanje podatkov** (angl. Data administration) – kadar si več uporabnikov deli podatke, lahko centralizirano upravljanje podatkov ponudi pomembne izboljšave. Izkušeni strokovnjaki, ki razumejo proces upravljanja podatkov in način, kako jih uporabljajo različne skupine uporabnikov, so lahko odgovorni za organizacijo predstavitve podatkov na način, da se zmanjša izguba in za dobro shranjevanje podatkov, da je njihova ponovna pridobitev učinkovitejša.
- **Sočasni dostop in obnovev zrušitve** (angl. Concurrent access and crash recovery) – DBMS omogoča sočasne dostope do podatkov na način, da lahko uporabniki mislijo, da do podatkov dostopa samo en uporabnik naenkrat. Poleg tega DBMS ščiti uporabnike pred učinki sistemskih napak.

- **Zmanjšan čas razvoja aplikacij** (angl. Reduced application development time) – DBMS podpira številne pomembne funkcije, ki so skupne mnogim aplikacijam, ki dostopajo do podatkov, shranjenih v DBMS. To skupaj v povezavi z vmesnikom višje ravni omogoča hiter razvoj aplikacij.

### **3 PRIMERJAVA SISTEMOV ZA UPRAVLJANJE S PODATKOVNIMI BAZAMI**

#### **3.1 MySQL**

MySQL je odprtokodni relacijski DBMS. Odprtokodna baza podatkov je vsaka aplikacija baze podatkov, ki si jo lahko uporabniki brezplačno ogledajo, prenesejo, spremenijo in ponovno uporabijo (Pure Storage, Inc., brez datuma). Poznan je kot eden izmed najbolj zanesljivih in najučinkovitejših DBMS. Ima možnost organiziranja podatkov v eno ali več podatkovnih tabel, v katerih so tipi podatkov med seboj povezani. Kot druge relacijske zbirke podatkov tudi MySQL shranjuje podatke v tabele, sestavljene iz vrstic in stolpcev. Uporabniki lahko definirajo, manipulirajo, nadzorujejo in poizvedujejo po podatkih z uporabo jezika SQL (Drake, 2020).

Uporabniki se ob uporabi sistema MySQL srečujejo z nižjimi stroški skupnega lastništva. Ker je odprtokoden, ga lahko uporabljajo brezplačno in lahko po želji njegovo izvorno kodo prilagodijo svojim zahtevam. Zanj je značilna prenosljivost kar pomeni, da lahko deluje na različnih platformah, kot so npr. Linux, Solaris in Windows (Munasingha, 2021). Je dobra izbira za projekte, ki ciljajo na več platform, saj je del sklada strežnikov Linux, Apache, MySQL in PHP, ki se uporabljajo po celem svetu za razvoj spletnih aplikacij (Pomponio, 2021). Za povezavo s strežnikom MySQL so na voljo različni varni in brezhibni mehanizmi povezovanja. MySQL ima veliko skupnost razvijalcev, ki izdaja redne popravke in posodobitve ter zagotavlja neprekinjeno delovanje in ponuja široko paleto rešitev visoke razpoložljivosti (Munasingha, 2021).

#### **3.2 Oracle Database**

Oracle Database družbe Oracle Corporation je relacijski DBMS. Prvotno razvit Oracle DB v letu 1977 je eden izmed najbolj zaupanja vrednih in najpogosteje uporabljenih podatkovnih motorjev relacijskih baz podatkov za shranjevanje, organiziranje in pridobivanje podatkov po vrsti, hkrati pa ohranja relacije med različnimi vrstami (Buttice, 2021). Ima sheme in tabele, tako kot večina drugih baz podatkov, vendar je njegova struktura optimizirana za najboljše delovanje s programsko opremo Oracle. Takšna programska oprema vključuje sistem Oracle HRMS, sistem Oracle Manufacturing ERP in sistem Oracle WebCenter Content (Computer Hope, 2017).

Po želji lahko uporabniki v zbirko podatkov Oracle dodajo dodatne tabele, tabelam v bazi podatkov pa dodatne stolpce. Za dostop do podatkov v bazi podatkov Oracle je potrebna uporaba programskega jezika SQL. Sintaksa SQL za Oracle je nekoliko drugačna od sintakse za druge vrste baz podatkov, splošna struktura "Oracle SQL" pa je relativno enaka sintaksi SQL za druge baze podatkov (Computer Hope, 2017).

Velikost baze podatkov Oracle se razlikuje glede na njeno uporabo, vendar lahko postane zelo velika, kar pomeni na stotine gigabajtov ali celo terabajtov. Za uporabo je potrebna licenca podjetja Oracle. Stroški licenciranja se razlikujejo glede na primer uporabe. Zaradi visokih stroškov Oracle pogosteje uporabljajo večja podjetja (Computer Hope, 2017).

### **3.3 Microsoft SQL Server**

SQL Server je priljubljen relacijski DBMS, ki ga je razvil Microsoft. SQL Server primarno shranjuje in pridobiva podatke, ki jih zahtevajo druge aplikacije. Te aplikacije se lahko izvajajo na istem ali drugem računalniku. Prvič je bil predstavljen leta 1989. Skozi leta se širi z dodajanjem novih funkcij. Večinoma se uporablja za obsežne poslovne aplikacije (Satapathi, 2021). Obstaja veliko različic Microsoft SQL Server-ja, ki skrbijo za različne delovne obremenitve in zahteve. Različica podatkovnega centra je prilagojena višji ravni podpore aplikacij in razširljivosti, medtem ko je različica Express pomanjšana, brezplačna izdaja programske opreme (Burns, 2016).

Microsoft SQL Server ima številne aplikacije v poslovnem svetu. Prva in najbolj očitna je, da se baza podatkov uporablja za shranjevanje in upravljanje informacij. Vendar pa bodo podjetja, ki hranijo občutljive podatke o strankah, kot so osebni podatki, podatki o kreditnih karticah in drugi zaupni podatki, imela koristi od povečane varnosti. Sistem omogoča tudi skupno rabo podatkovnih datotek z računalniki v istem omrežju, kar je povečalo zanesljivost. SQL Server se uporablja tudi za povečanje hitrosti obdelave podatkov, kar omogoča enostavno izvajanje velikih operacij. Z informacijami, shranjenimi v bazi podatkov imajo podjetja zanesljiv sistem varnostnega kopiranja (Infotec, 2017).

### **3.4 IBM DB2**

DB2 je relacijski DBMS. Zasnovan je za učinkovito shranjevanje, analizo in pridobivanje podatkov. Organizira shranjevanje velikega nabora različnih vrst podatkov in optimizira kasnejše pridobivanje le teh. Izvaja standardne koncepte relacijske baze, standardni jezik je SQL. Služi lahko kot tradicionalna relacijska baza, hierarhična, objektno usmerjena baza podatkov XML, shramba za JSON ali shramba ključ – vrednost, odvisno od implementacije (Crooks, 2018).

DB2 obstaja že od začetka relacijske baze podatkov, vendar se njegove funkcije redno posodablja. V DB2 je vložena ogromno inženiringa. DB2 ni odprtokoden, ampak ima nekaj brezplačnih distribucij. Če ga uporabniki želijo izvajati na večjih strežnikih, morajo



plačati ustrezno licenciranje. V celoti je v lasti IBM-a. Uporablja se lahko na različnih platformah, kot so Db2 za z/OS, Db2 za iSeries in Db2 za Linux, Unix in Windows. Zdaj deluje tudi na Dockerju (Crooks, 2018).

### **3.5 Microsoft Access**

Microsoft Access je psevdo-relacijski DBMS. To pomeni, da vsebuje določene elemente relacijske baze podatkov. Je del zbirke aplikacij Microsoft Office, ki med drugimi vključuje tudi Word, Outlook in Excel. Ravno tako je na voljo kot samostojen izdelek. Uporablja se za uvajanje majhnih in velikih podatkovnih baz, kar je posledica njegovega enostavnega grafičnega vmesnika in njegove združljivosti z drugimi aplikacijami in platformami, kot sta Microsoftov lastni DBMS SQL Server in Visual Basic za aplikacije (angl. Visual Basic for Applications – VBA) (Techopedia, 2016).

Access uporabnikom omogoča ustvarjanje podatkovnih baz po meri, ki hrani informacije v organizirani strukturi. Program ponuja tudi vizualni vmesnik za ustvarjanje prilagojenih obrazcev, tabel in poizvedb SQL. Podatke je mogoče vnesti v Accessovo bazo podatkov z vizualnimi obrazci ali z osnovnim vmesnikom za preglednice. Po informacijah, shranjenih v podatkovni bazi Access, je mogoče iskati in dostopati tudi iz drugih programov, vključno s spletnimi storitvami (PC.net, brez datuma).

### **3.6 PostgreSQL**

PostgreSQL ali Postgres je odprtokodni, objektno-relacijski DBMS, ki ni v lasti enega podjetja ali posameznika. Ustvarjen je bil s ciljem, da je zelo razširljiv in skladen z aktualnimi standardi. Čeprav je predvsem relacijska baza podatkov, vključuje tudi funkcije, kot sta dedovanje tabel in preobremenitev funkcij, ki so pogosteje povezane z objektnimi bazami podatkov (Drake, 2014).

Prislužil si je močan sloves s svojo dokazano arhitekturo, zanesljivostjo, celovitostjo podatkov, robustnim naborom funkcij, razširljivostjo in predanostjo skupnosti odprte kode, ki stoji za programsko opremo, da dosledno zagotavlja zmogljive in inovativne rešitve. PostgreSQL deluje na vseh večjih operacijskih sistemih, je združljiv z ACID od leta 2001 in ima zmogljive dodatke, kot je priljubljena razširitev geoprostorske baze podatkov PostGIS. Uporablja in razširja jezik SQL v kombinaciji s številnimi funkcijami, ki varno shranjujejo in spreminjajo najbolj zapletene delovne obremenitve podatkov (PostgreSQL, brez datuma).

### **3.7 Primerjava DBMS**

Pri primerjavi sistemov sem si pomagala s štirimi tabelami. V tabeli 1 so predstavljene osnovne informacije DBMS.

*Tabela 1: Osnovne informacije DBMS*

	<b>Vrsta sistema</b>	<b>Razvoj</b>	<b>Prva izdaja</b>	<b>Licenca</b>	<b>Model podatkovne baze</b>
MySQL	Relacijski	Sun Microsystems (kasneje Oracle Corporation)	1995	Splošno dovoljenje GNU ali lastniška	Relacijski
Oracle	Relacijski	Oracle Corporation	1980	Lastniška	Relacijski
MS SQL	Relacijski	Microsoft	1989	Lastniška	Relacijski
IBM Db2	Relacijski	IBM	1983	Lastniška	Relacijski
Microsoft Access	Psevdo-relacijski	Microsoft	1992	Lastniška	Relacijski
PostgreSQL	Objektno relacijski	PostgreSQL Global Development Group	1989	Postgres licenca	Relacijski

*Vir: lastno delo.*

Tabela 2 prikazuje podprte operacijske sisteme DBMS.

*Tabela 2: Podprti operacijski sistemi DBMS*

	<b>Windows</b>	<b>macOS</b>	<b>Linux</b>	<b>UNIX</b>	<b>z/OS</b>	<b>iOS</b>	<b>Android</b>
MySQL	DA	DA	DA	DA	DA	NE	DA
Oracle	DA	DA	DA	DA	DA	NE	NE
MS SQL	DA	NE	DA	NE	DA	NE	NE
IBM Db2	DA	DA	DA	DA	DA	DA	NE
Microsoft Access	DA	NE	NE	NE	NE	NE	NE
PostgreSQL	DA	DA	DA	DA	NE	NE	DA

*Vir: lastno delo.*

V tabeli 3 najdemo temeljne značilnosti DBMS.

*Tabela 3: Temeljne značilnosti DBMS*

	<b>ACID</b>	<b>Referenčna integriteta</b>	<b>Transakcije</b>	<b>Vmesnik</b>
MySQL	DA	DA	DA, razen za DDL	GUI in SQL
Oracle	DA	DA	DA, razen za DDL	API, GUI in SQL
MS SQL	DA	DA	DA	GUI in SQL
IBM Db2	DA	DA	DA	GUI in SQL

se nadaljuje

Tabela 3: Temeljne značilnosti DBMS (nad.)

Microsoft Access	DA	DA	DA	GUI in SQL
PostgreSQL	DA	DA	DA	API, GUI in SQL

Vir: lastno delo.

Štiri ključne lastnosti transakcij so atomičnost, doslednost, izolacija in trajnost (angl. atomicity, consistency, isolation, durability – ACID). So nabor načel, ki zagotavljajo zanesljivo obdelavo transakcij podatkovne baze (Watts, 2022).

Referenčna integriteta je koncept relacijske baze podatkov, ki narekuje, da morajo biti povezave v tabeli konsistentne. Spremembe glavnega ključa morajo biti upoštevane pri vseh tujih ključih. Preverja se, ko se podatki v tabelo vstavljajo, popravljajo ali brišejo (IBM Corporation, 2021).

Transakcije so niz operacij, ki se uporabljajo za izvajanje nekega logičnega sklopa dela. Za razliko od podatkov v bazi podatkov, se transakcija izvede, kar je mogoče storiti z vstavljanjem novih podatkov, posodobitvijo obstoječih ali z brisanjem (Singh, 2022).

Tabela 4 prikazuje omejitve DBMS.

Tabela 4: Omejitve DBMS

	Maksimalna velikost podatkovne baze	Maksimalna velikost tabele	Maksimalna velikost ene vrstice	Maksimalno število stolpcev na vrstico
MySQL	Neomejeno	65,536 TB	65,535 KB	4096
Oracle	128 TB	65,536 TB	največ 4 GB, odvisno od podatkovnega tipa	1000
MS SQL	524,272 TB	524,272 TB	8060 B/2 TB	1024/30000
IBM Db2	Neomejeno	64 TB	1048,319 KB	1012
Microsoft Access	2 GB	2 GB	16 MB	255
PostgreSQL	Neomejeno	32 TB	1,6 TB	1600

Vir: lastno delo.

## 4 DISKUSIJA

Parsons (2018) je prepričana, da zaradi hitrih sprememb v tehnologiji poslovne organizacije ne delujejo več na način, kot so nekoč. Mobilni telefoni, internet in socialni mediji predstavljajo nekaj nedavnih tehnologij, ki niso spremenile le družbe, ampak tudi

način poslovanja in interakcije s strankami in zaposlenimi. Podjetja so se bila prisiljena prilagajati času, ena od teh posvojitvev pa je uvedba DBMS. Ti sistemi omogočajo poslovnim organizacijam shranjevanje, spreminjanje in pridobivanje informacij iz baz podatkov. Uporabnikom in strankam so omogočili integracijo na načine, ki še nikoli niso obstajali, ter omogočili, da se informacije delijo in raziščejo bolj temeljito in jasneje. Na splošno obstajajo številne prednosti DBMS, ki so poslovnim organizacijam omogočile, da resnično napredujejo.

Ena od glavnih poslovnih prednosti je povečanje zmožnosti spremljanja in razvrščanja pomembnih podatkov. Podjetja lahko v svojih bazah podatkov shranijo informacije, ki se nanašajo na stranke, prodajalce, zaposlene, plače, zaloge, račune, naročila in drugo. DBMS omogočajo večjo rast podjetja, saj olajšajo shranjevanje velikih količin podatkov in s tem upravljanje večjega števila strank brez zmanjšanja pričakovanj glede storitev. S papirnatimi datotekami se informacije lažje izgubijo, kar vodi do druge pomembne poslovne prednosti sistemov za upravljanje baz podatkov: organizacija in natančnost. Ko so vsi podatki v spletni bazi podatkov, je veliko lažje hitro pridobiti informacije, za katere je skoraj zagotovljena točnost. To pomeni več doslednosti in manj verjetnosti, da bi prišlo do človeških napak (Parsons, 2018).

DBMS ne pomaga le pri učinkovitem shranjevanju podatkov, temveč tudi pri izmenjavi podatkov znotraj organizacije. Ljudje, vključeni v takšno okolje so programerji, sistemski skrbniki in končni uporabniki. V okolju baze podatkov je pet komponent: podatki, strojna oprema, programska oprema, ljudje in postopki. Več uporabnikov ima lahko hiter dostop do podatkov in jih deli z drugimi uporabniki brez dodatnega napora. Sistem pomaga pri iskanju zahtevanih informacij iz velike baze podatkov v samo nekaj sekundah. Zato je za podjetja zelo učinkovito, da upravljajo različne vrste podatkov, kot so evidence študentov, podatki o zaposlenih, inventar, obračun plač, vodenje projektov in drugo (Wagener, 2021).

Organizacijsko strukturo podjetja lahko razdelimo na tri ravni: najvišjo, srednjo in operativno. Podatkovni sistem mora zagotoviti orodja, ki dajejo vsaki izmed naštetih ravni primeren pogled na podatke in podpirajo zahtevano raven odločanja. Najvišje vodstvo sprejema strateške odločitve, zato mora ustrezen DBMS zagotavljati informacije, potrebne za strateško odločanje, načrtovanje, oblikovanje politike in opredelitve ciljev. Srednje vodstvo sprejema taktične odločitve, kjer spremljajo in nadzirajo dodeljevanje in uporabo sredstev podjetja ter uspešnost oddelkov. Operativno vodstvo pa potrebuje ustrezen DBMS za predstavljanje in podpiranje poslovanja podjetja. Sistem mora biti zmožen izdelati rezultate poizvedb v okviru pričakovanih zmogljivosti. Baza podatkov mora podpirati hitre odzive na večje število transakcij na ravni operativnega upravljanja. Podatkovni model mora biti dovolj prilagodljiv, da vključuje vse zahtevane sedanje in pričakovane podatke (Thiru, brez datuma).

Zaključna strokovna naloga ima tudi nekaj omejitev. Med pisanjem in iskanjem podatkov sem naletela na nekaj izzivov. Na prve težave sem naletela že pri izbiri literature. Zaradi

obilice in raznolikosti tako knjižnih kot internetnih virov, je bilo na trenutke težko presoditi, kateri viri so najbolj verodostojni in bistveni. Medtem ko je bilo za nekatere DBMS na voljo ogromno literature, je bilo za pridobitev določenih podatkov o nekaterih drugih sistemih na voljo le malo ali nič literature. Če bi želela posamezne sisteme podrobneje analizirati, bi presegla obseg zaključne naloge, ki znaša 20–25 strani, zato so na nekaterih mestih podrobnosti izpuščene. Da bi lahko DBMS podrobneje analizirala in primerjala, bi se morala z njimi neposredno seznaniti in jih dalj časa uporabljati, kar pa v okviru izdelave zaključne naloge ni bilo izvedljivo.

#### **4.1 Smernice za nadaljnje delo**

Za boljšo primerjavo DBMS bi bilo dobro pod drobnogled vzeti tiste, ki so trenutno na trgu najbolj popularni. Med naštete bi lahko vključila tudi nerelacijska, odprtokodna DBMS MongoDB in Redis, ki po Statista (2022a) spadata na lestvico prvih šestih najpopularnejših DBMS na trgu v letu 2022.

Za čimbolj natančno analizo bi bilo smiselno sisteme grupirati po vrsti in modelu, nato pa iz vsakega dobljenega clustra izbrati izstopajočega po opazovanih kriterijih. Možno bi bilo izvesti tudi intervjuje z uporabniki opazovanih DBMS in primerjati rezultate intervjujev z rezultati naloge.

Med kriterije omejitev DBMS v tabeli 4 bi bilo možno vključiti maksimalno CHAR velikost (podatkovni tip za shranjevanje črk), maksimalno velikost števila, minimalno in maksimalno vrednost datuma ter maksimalno velikost naslova stolpca. Smiselno bi bilo raziskati, če imajo DBMS možnost začasnih tabel (angl. Temporary table) in pogledov (angl. View), kakšne zmožnosti ima podatkovna baza pri izvajanju ukazov s pomočjo poizvedovalnega jezika in katere podatkovne tipe pozna posamezni DBMS. Predstaviti bi bilo mogoče tudi zgradbe oziroma arhitekture sistemov. Na sisteme DBMS bi lahko gledali tudi z vidika cen in kasnejših stroškov nadgrajevanj.

### **SKLEP**

Z obravnavo teme v zaključni nalogi sem želela pritegniti pozornost k podatkovnim bazam in DBMS. Dejstvo je, da živimo v vedno bolj povezanem svetu, kjer se informacije pretakajo med nami, organizacijami in podjetji, s katerimi se srečujemo vsak dan. Količina podatkov narašča, sorazmerno pa naraščajo potrebe po novih podatkovnih bazah, ki lahko hitreje in učinkovitejše upravljajo z njimi. V nalogi sem opisala relacijski, hierarhični, mrežni, objektni, večdimenzionalni in stolpčno usmerjeni podatkovni model ter NoSQL podatkovno bazo.

Razširljivost DBMS se že dolgo raziskuje tako v akademskih krogih kot industriji. Ker je danes mogoče izbirati med številnimi DBMS, lahko izbiranje ene rešitve pred drugo

postane zapletena odločitev. V delu sem se primarno osredotočila na primerjavo šestih DBMS. Pomagala sem si s štirimi tabelami: tabela 1 zajema osnovne informacije sistemov, iz tabele 2 je razvidno katere podprte sisteme ima posamezen DBMS, v tabeli 3 so predstavljene temeljne značilnosti sistemov, v tabeli 4 pa omejitve oziroma zmogljivosti DBMS.

Po podatkih Statista (2022a) so od januarja najbolj priljubljena vrsta relacijski DBMS z 72,2 odstotka skupnih ocen razvrstitve sistemov. Najbolj priljubljen DBMS na svetu je bil komercialni sistem Oracle, sledil mu je odprtokodni sistem MySQL in komercialni sistem Microsoft SQL Server. Številni DBMS so brezplačne, odprtokodne tehnologije, kot sta MySQL in PostgreSQL, vendar imajo pri oblikovanju industrije pomembno vlogo tudi velika podjetja, kot so Microsoft, Oracle in IBM. 79,8 odstotka ocen sistemom za upravljanje baz podatkov je bilo namenjenim tistim odprtokodnih licenc (Statista, 2022b).

Med drugimi je bil poudarek tudi na pomenu ugotovitev za podjetja. Temeljno vprašanje, ki sem mu sledila, je bilo, kakšne koristi imajo podjetja od podatkovnih baz oziroma DBMS. Prišla sem do sklepa, da poleg tehničnih prednosti DBMS za več vrst organizacij obstajajo tudi posebne prednosti za poslovanje. Ti sistemi so integrirani v vmesnik, ki ga stranke uporabljajo za nakupe v podjetju. Funkciji razvrščanja podatkov in iskanja strankam olajšajo preprosto iskanje po podatkovni bazi podjetja s pomočjo spletnega mesta. Ponujena jim je možnost iskanja izdelkov po različnih kriterijih, kot sta cena in blagovna znamka ter opravilo nakupa brez težav. DBMS so še posebej uporabni za podjetja, ki prodajajo svoje izdelke javnosti.

Danes imajo ljudje zaradi tehnologije, kot so DBMS, večja pričakovanja glede poslovnih transakcij. Zaradi brskalnikov, kot je Google, so navajeni imeti vse željene informacije takoj na dosegu roke. Človeške interakcije naj bi bile hitre, kratke in jasne. In kot pravi Parsons (2018), trenutna doba je resnično doba podatkov in informacij.

## LITERATURA IN VIRI

1. Amazon Web Services, Inc. (brez datuma). *What is NoSQL?* Pridobljeno 14. maja 2022 iz <https://aws.amazon.com/nosql/>
2. BBC. (brez datuma). *Introducing databases.* Pridobljeno 17. maja 2022 iz <https://www.bbc.co.uk/bitesize/guides/zfd2fg8/revision/1>
3. Burns, E. (2016, 6. junij). *What is SQL Server?* Pridobljeno 24. maja 2022 iz <https://techmonitor.ai/what-is/what-is-sql-server-4914415>
4. Buttice, C. (2021, 21. junij). *Oracle Database (Oracle DB).* Pridobljeno 20. maja 2022 iz <https://www.techopedia.com/definition/8711/oracle-database>
5. C# Corner. (2019, 6. september). *What Are Object-Oriented Databases And Their Advantages.* Pridobljeno 5. junija 2022 iz <https://www.c-sharpcorner.com/article/what-are-object-oriented-databases-and-their-advantages2/>

6. Chauhan, K., Kumar, A. & Kohli, A. (2017, november). Evolution and development of dbms in software development industry. *The International Journal of Emerging technologies and Innovative Research*, 4(11), 538–542.
7. Computer Business Research. (brez datuma). *Network database model*. Pridobljeno 14. maja 2022 iz <https://www.computerbusinessresearch.com/Home/database/network-database-model/>
8. Computer Hope. (2017, 10. februar). *Oracle database*. Pridobljeno 20. maja 2022 iz <https://www.computerhope.com/jargon/o/oracle-database.htm>
9. Crooks, E. (2018, 6. marec). *Db2 Basics: Introduction to IBM Db2* [objava na blogu]. Pridobljeno 14. junija 2022 iz <https://datageek.blog/en/2018/03/06/db2-basics-introduction-to-ibm-db2/>
10. Drake, M. (2014, 21. februar). *SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems*. Pridobljeno 12. junija 2022 iz <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
11. Drake, M. (2020, 14. december). *What is MySQL?* Pridobljeno 20. maja 2022 iz <https://www.digitalocean.com/community/tutorials/what-is-mysql>
12. Elmasri, R. & Navathe, S. (2007). *Fundamentals of database systems* (4. izd.). Boston: Pearson Addison-Wesley.
13. European Business Review. (2020, 27. september). *How Databases Can Help Improve Business Performance*. Pridobljeno 11. aprila 2022 iz <https://www.europeanbusinessreview.com/how-databases-can-help-improve-business-performance/>
14. Foote, K. D. (2018, 19. junij). *A Brief History of Non-Relational Databases*. Pridobljeno 5. junija 2022 iz <https://www.dataversity.net/a-brief-history-of-non-relational-databases/>
15. Garcia-Molina, H., Ullman, J. D. & Widom, J. (2008). *DATABASE SYSTEMS - The Complete Book* (2. izd.). Upper Saddle River, New Jersey: Pearson Prentice Hall.
16. Geeks for Geeks. (2021, 3. februar). *What is database?* Pridobljeno 11. aprila 2022 iz <https://www.geeksforgeeks.org/what-is-database/>
17. Geeks for Geeks. (brez datuma). *Difference between Open Source Database and Commercial Database*. Pridobljeno 5. junij 2022 iz <https://www.geeksforgeeks.org/difference-between-open-source-database-and-commercial-database/>
18. Gillis, A. S. (brez datuma). *Multidimensional database (MDB)*. Pridobljeno 28. aprila 2022 iz <https://www.techtarget.com/searchoracle/definition/multidimensional-database>
19. Graham, A. (brez datuma). *How Are Databases Used in the Real World?* Pridobljeno 24. junija 2022 iz <https://www.techwalla.com/articles/how-are-databases-used-in-the-real-world>
20. Gunjal, B. (2003). Database System: Concepts and Design. V *Proceedings of 24th IASLIC-SIG-2003* (str. 1-20). Sundargarh: National Institute of Technology Rourkela.

21. Heavy.ai. (brez datuma). *Hierarchical Database*. Pridobljeno 11. aprila 2022 iz <https://www.heavy.ai/technical-glossary/hierarchical-database>
22. Hill, J. (2021, 15. junij). *Data vs Information: What's the Difference?* [objava na blogu]. Pridobljeno 5. julija 2022 iz <https://bloomfire.com/blog/data-vs-information/>
23. IBM Cloud Education. (2019, 6. avgust). *Relational Databases*. Pridobljeno 12. aprila 2022 iz <https://www.ibm.com/cloud/learn/relational-databases>
24. IBM Corporation. (2021, 1. marec). *Referential integrity*. Pridobljeno 16. julija 2022 iz <https://www.ibm.com/docs/en/informix-servers/14.10?topic=integrity-referential>
25. Infotec. (2017, 17. julij). *What is Microsoft SQL Server and What is it Used For?* [objava na blogu]. Pridobljeno 24. maja 2022 iz <https://www.infotctraining.com/blog/what-is-microsoft-sql-server-and-what-is-it-used-for>
26. Kozubek-Krycuń, A. (2022, 8. december). *The History of SQL Standards* [objava na blogu]. Pridobljeno 12. aprila 2022 iz <https://learnsql.com/blog/history-of-sql-standards/>
27. Kumar, D. (2021, 22. junij). *What is Columnar Database? – A Comprehensive Guide 101*. Pridobljeno 9. maja 2022 iz <https://hevodata.com/learn/columnar-databases/#cdb>
28. Microsoft. (brez datuma). *What are NoSQL databases?* Pridobljeno 14. maja 2022 iz <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-nosql-database/>
29. MongoDB, Inc. (brez datuma). *What Is an Object-Oriented Database?* Pridobljeno 14. maja 2022 iz <https://www.mongodb.com/databases/what-is-an-object-oriented-database>
30. Munasingha, D. (2021, 21. maj). *Advantages and Disadvantages of using MySQL*. Pridobljeno 23. maja 2022 iz <https://diliru.medium.com/advantages-and-disadvantages-of-using-mysql-36f6ffce3fa3>
31. Oracle. (brez datuma a). *What is database?* Pridobljeno 11. aprila 2022 iz <https://www.oracle.com/database/what-is-database/>
32. Oracle. (brez datuma b). *What is a Relational Database (RDBMS)?* Pridobljeno 11. aprila 2022 iz <https://www.oracle.com/database/what-is-a-relational-database/>
33. Parsons, G. (2018, 28. januar). *Importance of Database Management Systems*, str. 1-4. Pridobljeno 15. junija 2022 iz <https://www.studocu.com/en-us/document/st-johns-university/database-management/why-database-management-systems-are-important-to-business-organizations/1757836>
34. PC.net. (brez datuma). *Access*. Pridobljeno 30. maja 2022 iz <https://pc.net/glossary/definition/access>
35. Pedamkar, P. (brez datuma). *Multidimensional Database*. Pridobljeno 28. aprila 2022 iz <https://www.educba.com/multidimensional-database/>
36. Pomponio, A. (2021, 14. oktober). *MySQL Overview: Key Features, Benefits, and Use Cases* [objava na blogu]. Pridobljeno 20. maja 2022 iz <https://www.openlogic.com/blog/mysql-overview>



37. PostgreSQL. (brez datuma). *What is PostgreSQL?* Pridobljeno 24. junija 2022 iz <https://www.postgresql.org/about/>
38. Pure Storage, Inc. (brez datuma). *What Is an Open Source Database?* Pridobljeno 20. maja 2022 iz <https://www.purestorage.com/knowledge/what-is-an-open-source-database.html>
39. Quickbase. (brez datuma). *A Timeline of Database History & Database Management.* Pridobljeno 11. aprila 2022 iz <https://www.quickbase.com/articles/timeline-of-database-history>
40. Ramakrishnan, R. & Gehrke, J. (2011). *Database Management Systems.* New York: McGraw-Hill Higher Education.
41. Satapathi, A. (2021, 20. september). *Understanding Database Management Systems.* Pridobljeno 24. maja 2022 iz <https://www.c-sharpcorner.com/article/understanding-database-management-systems/>
42. ScyllaDB. (brez datuma). *Columnar Database.* Pridobljeno 9. maja 2022 iz <https://www.scylladb.com/glossary/columnar-database>
43. Singh, N. (2022, 25. januar). *What is Transaction in DBMS?* Pridobljeno 5. junija 2022 iz <https://www.scaler.com/topics/dbms/transaction-in-dbms/>
44. Slv.kagutech.com. (2022). *Hierarhična baza podatkov je ... Modeli, primeri.* Pridobljeno 11. julija 2022 iz <https://slv.kagutech.com/4108058-the-hierarchical-database-is-...-models-examples>
45. Statista. (2022a, 23. maj). *Popularity breakdown of database management systems (DBMSs) worldwide as of January 2022, by license/category.* Pridobljeno 24. junija 2022 iz <https://www.statista.com/statistics/1132409/worldwide-popularity-database-management-systems-category-license/>
46. Statista. (2022b, 23. maj). *Popularity comparison of database management systems (DBMSs) worldwide as of January 2022, by category.* Pridobljeno 24. junija 2022 iz <https://www.statista.com/statistics/1131595/worldwide-popularity-database-management-systems-category/>
47. Sumathi, S. & Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems* (47. izd.). Berlin Heidelberg: Springer.
48. Tech Target Contributor. (2021). *Columnar Database.* Pridobljeno 9. maja 2022 iz <https://www.techtarget.com/searchdatamanagement/definition/columnar-database>
49. Techopedia. (2016, 28. december). *Microsoft Access.* Pridobljeno 30. maja 2022 iz <https://www.techopedia.com/definition/1218/microsoft-access>
50. Techopedia. (2020, 30. junij). *Semantic Data Model.* Pridobljeno 30. maja 2022 iz <https://www.techopedia.com/definition/30489/semantic-data-model>
51. Think Automation. (brez datuma). *The History of Databases.* Pridobljeno 11. aprila 2022 iz <https://www.thinkautomation.com/histories/the-history-of-databases/>
52. Thiru. (brez datuma). *Role of Databases in An Organizations.* Pridobljeno 22. junija 2022 iz <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/569-role-of-databases-in-an-organizations.html>

53. Wagener, A. (2021, 23. februar). *What is The Importance of a Database Management System?* Pridobljeno 15. junija 2022 iz <https://www.mcg.dk/post/what-is-the-importance-of-a-database-management-system>
54. Watt, A. (brez datuma). *Chapter 3: Characteristics and Benefits of a Database.* Pridobljeno 17. maja 2022 iz <https://opentextbc.ca/dbdesign01/chapter/chapter-3-characteristics-and-benefits-of-a-database/>
55. Watts, S. (2022, 24. april). *ACID Explained: Atomic, Consistent, Isolated & Durable* [objava na blogu]. Pridobljeno 5. junija 2022 iz <https://www.bmc.com/blogs/acid-atomic-consistent-isolated-durable/>