**Unit II**

**Sistem za rukovođenje bazom podataka**

Kompjuteru, baza podataka izgleda kao jedan ili više datoteka. Kako bi podaci iz baze podataka bili pročitani, izmenjeni, dodati ili uklonjeni, softverski program mora da dopre do njih (da ima pristup). Mnoge softverske aplikacije imaju ovu sposobnost. iTunes može da pročita sopstvenu bazu podataka i pruži listu pesama (i reprodukuje ih (pesme)); Vaš softver na mobilnom telefonu može da interaguje sa Vašom listom kontakata.

Ali šta je sa aplikacijama koje treba da kreiraju i upravljaju bazom podataka? Koji softver možete da koristite kako bi kreirali bazu podataka, menjali njenu strukturu (sastav) ili jednostavno uradili analizu iste? Ovo je svrha cele kategorije softverskih aplikacija nazvanih sistemima upravljanja bazom podataka (DBMS) lol.

Sistem za upravljanje bazom podataka omogućava svom korisniku da napravi, pročita, ažurira i obriše podatke iz baze podataka. Sistem za upravljanje bazom podataka u svojoj osnovi služi kao interfejs između baze podataka i korisnika ili programskih aplikacija, osiguravajući da su podaci dosledno organizovani i ostaju lako dostupni. DBMS-a upravlja (rukovodi, kontroliše, nadgleda) tri bitne stvari: podatke, pokretač (motor) baze podataka koji dozvoljava da se podacima pristupa, da se oni zaključavaju i da se nad njima vrše izmene, i shemu baze podataka, koja definiše logičku strukturu baze podataka. Ova tri osnovna elementa (DBMS-a) pomažu pri omogućavanju istovremenog pristupa (podudarnosti), bezbednosti, integritetu podataka i uniformnosti (jednoličnosti, ujednačenosti, homogenosti) administrativnim procedurama. Tipični administrativni zadaci baze podataka podržani od strane DBMS-a uključuju: zamenu u kontroli, nadgledanje, štelovanje performansi, čuvanje rezervne kopije i obnovu baze podataka. Mnogi sistemi za kontrolu nad bazom podataka su takođe odgovorni za automatski povratak, restartovanje (ponovno pokretanje) i obnavljanje kao i beleženje i reviziju aktivnosti (pri pristupu bazi).

Ključna komponenta Sistema za kontrolu baze podataka je pokretač (motor) baze podataka – deo programa koji zapravo skladišti i preuzima podatke. Pored motora baze podataka, većina DBMS-a dolaze upakovane sa skupom alata kojima se mogu izvršiti različiti neophodni poslovi (zadaci), poput formi za kreiranje (koriste se za unos podataka) i izveštaji (koji se koriste za prikaz rezultata), kao i međusobno povezivanje sa upitnim i programerskim jezicima za rad sa kompleksnim aplikacijama. Programski jezici koji se uglavnom koriste sa bazama podataka podrazumevaju Visual Basic, Java, C++, iako mnogi stariji nasleđeni sistemi baza podataka i dalje koriste COBOL.

DBMS paketi obično omogućavaju interfejs za pregled i izmenu dizajna baze podataka, za kreiranje upita i razvoja izveštaja. Većina ovih paketa je dizajnirana (osmišljena) da radi sa određenim (preciziranim) tipovima baze podataka, ali su obično kompatibilne (slažu se, usklađene) sa velikim opsegom baza podataka.

Na primer, Apache OpenOffice.org Base može da se koristi za kreiranje, izmenu i analizu baza podataka u otvorenom formatu baze podataka. Microsoft-ov Access DBMS se koristi za rad sa bazama podataka u njegovom sopstvenom Microsoft Access Database formatu. Oba, i Access i Base takođe imaju mogućnost (sposobnost) da čitaju i pišu na druge (u drugim) formate baze podataka.

Microsoft Access i Open Office Base su primeri ličnih sistema za upravljanje bazama podataka. Ovi sistemi su prvobitno korišćeni za ravoj i analizu jednokorisničkih baza podataka. Nije zamišljeno da se ove baze podataka dele preko mreža ili Interneta, već su instalirane na određeni uređaj i rade sa jednim korisnikom u jednom trenutku.

Sistemi za upravljanje bazama podataka su dizajnirani za mnoštvo različitih okruženja. Neki su osmišljeni kao jednokorisnički sistemi, dok su drugi dizajnirani za preduzeća srednje veličine, dok su još drugi (još postoji idk) dizajnirani za velika preduzeća.

**Prednosti DBMS-a**

Korišćenje DBMS-a za skladištenje i kontrolu podataka dolazi sa prednostima, ali i sa troškovima. Jedna od većih prednosti korišćenja DBMS-a jeste ta da dozvoljava korisnicima i programerima aplikacija da pristupaju i koriste iste podatke dok se upravlja (nadgleda, prati, vodi računa o) integritet podataka. Podaci su bolje zaštićeni i kontrolisaniji kada mogu da se dele uz pomoć DBMS-a umesto kreiranja novih iteracija istih podataka skladištenim u novim fajlovima za svaku novu aplikaciju. DBMS omogućava centralno skladištenje za podatke kojima se može pristupati od strane većeg broja korisnika na kontrolisan način.

Centralno skladištenje i upravljanje podacima unutar DBMS-a omogućava:

* Apstrakciju podataka i njihovu nezavisnost
* Sigurnost podataka
* Mehanizam za zaključavanje pri istovremenom (konkurentnom) pristupu
* Efikasan rukovodilac koji može da balansira potrebe većeg broja aplikacija koje koriste iste podatke
* Sposobnost da se brzo (sa lakoćom) podaci oporave nakon grešaka i pada sistema, uključujući u to i sposobnost ponovnog pokretanja i obnovljivost
* Sposobnost jakog integriteta nad podacima
* Beleženje i reviziju aktivnosti
* Jednostavan pristup uz pomoć standardnog (onaj koji je uzor, norma) interfejsa programskih aplikacija
* Jednolične admnistratorske procedure za podatke

Još jedna prednost DBMS-a jeste da može da se koristi za predlaganje logične, strukturne organizacije podataka (nad podacima). DBMS-a pruža (dostavlja) ekonomski priraštaj (rast) pri obradi velike količine podataka jer je optimizovan (naštelovan) za takve operacije.

DBMS takođe može da pruži različite preglede (viđenja) za jednu istu shemu baze podataka. Pregled definiše podatke koje korisnik vidi i na koji način (kako) ih vidi. DBMS omogućava (pruža) nivo apstrakcije između konceptualne šeme (skica, plan) koja definiše logičku strukturu baze podataka i fizičke sheme koja opisuje datoteke, indekse(re) i druge fizičke mehanizme koje baza podataka koristi. Pri korišćenju DBMS-a sistemi mogu biti mnogo lakše izmenjeni ukoliko (kada) se potrebe kompanije (biznisa) promene. Nove vrste (kategorije) podataka mogu biti dodate u bazu bez remećenja postojećeg sistema i aplikacije mogu biti izolovane (nezavisne) od toga kako su podaci strukturirani (strukturisani?) i skladišteni.

Naravno, DBMS mora da izvršava dodatan posao kako bi omogućio ove prednosti, time donoseći troškove. DBMS će koristiti više memorije i (vremena) Centralne Procesorske Jedinice od prostog sistema za skladištenje datoteka. I, naravno, različiti tipovi (vrste) DBMS-ova će zahtevati druge tipove i nivoe sistemskih resursa.

**Use of Language Practice**

* **Match words (1-10) to their synonyms (a-j)**

1. Concurrency – (e) the ability of a database to allow multiple users to affect multiple transactions
2. Backup – (d) the procedure for making copies of data in case the original is lost or damaged
3. Rollback – (g) the process of restoring a database or program to a previously defined state, typically to recover form an error
4. Logging – (e) just a fancy word to define a process of writing down everything you do
5. Query Language – (h) a language for the specifiacation of procedures for the retrieval (and sometimes modification) of information from a database.
6. Overhead – (b) in computer science, it is generally considered any combination of excess or indirect computation time, memory, bandwidth, or other resources that are required to attain a particular goal
7. Iteration – (c) repetition of a mathematical or computational procedure applied to the result of a previous application
8. Economy of scale – (a) a proportionate saving in costs gained by an increased level of production

* **Use the words above to complete the following sentences. Consider correct grammar use.**

1. In database technologies, a ***ROLLBACK*** is an operation which returns the database to some previous state. They are important for database integrity, because they mean that the database can be restored to a clean copy even after erroneous operations are performed.
2. All small island developing states lack the ***ECONOMY OF SCALE*** to overcome their vulnerabilities on their own.
3. SXPath may be used as a ***QUERY LANGUAGE*** for an XML-base digital library.
4. ***ITERATION*** enables refinement of the work product through encouraging brief returns to previous steps.
5. Make a ***BACKUP*** of any work you do on the computer.
6. ***LOGGING*** is keeping a record of all data input, processes, data output, and final results in a program.
7. The ability to offer ***CONCURRENCY*** is unique to database.
8. For example, maintaining an audit trail might result in 10% ***OVERHEAD***, meaning that the program will run 10% slower when the audit trail is turned on.