Факултет инжењерских наука

Универзитета у Крагујевцу

Назив студијског програма: Машинско инжењерство

Ниво студија: Основне академске студије

Модул: Информатика у инжењерству

Предмет: Базе података

Број индекса: 17/2010

Александра Д Митровић

Концепти сигурности база података

завршни рад

Комисија за преглед и одбрану:

1. **Др Милан Ерић, проф. - ментор**
2. _____ Датум одбране:_____
3. _____ Оцена:_____

У оквиру овог завршног рада кандидат треба да:

Проучи литературу и презентира концепте сигурности база података. Целине завршног рада треба да обухвате уводна разматрања, проблеме сигурности, концепте заштите, сигурносне механизме и закључна разматрања.

Препоручена литература:

1. Лазаревић Б. ….: ***Базе података***, ФОН Београд, Београд 2003.
2. Павловић-Лажетић Г.: ***Основе релационих база података***, Математички факултет, Београд, 2000.
3. Ross Mistry, Stacia Misner: ***Introducing Microsoft SQL Server 2008***, Microsoft Press, Washington, 2010.
4. Michael Lee, Gentry Bieker: ***Mastering SQL Server 2008***, Wiley Publishing, Inc., Indianapolis, Indiana, 2009.

Крагујевац, 2013. ментор:

Др Милан Ерић, проф.

Изјављујем да сам овај завршни рад урадила самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

Александра Митровић 17/2010

Садржај

Резиме

У овом раду је објашњен веома заступљен проблем данас, а то је како заштитити базу података. Сигурност базе података се односи на заштиту базе података од неовлашћеног коришћења, модификовања, намерног оштећења или уништења података. У тексту који следи акценат ће бити стављен на заштиту базе података од оштећења или уништења на нивоу система управљања базом података, као и на опоравак базе података.

Кључне речи: база података, СУБП, шифровање, резервна копија, опоравак базе података.

Abstract

This work is explained very popular issue today, and that is to protect the database. Database security refers to the protection of databases from unauthorized use, modification, intentional damage or destruction of data. In the following the focus will be to protect the database from damage or destruction at the database management system, as well as the recovery of the database.

Keywords: database, DBMS, encryption, backup, database recovery.

Увод

База података (енг. *Database*) је организована збирка информација, типично у дигиталном облику, које се помоћу специјализованих програмских алата може уређивати, брисати и додавати. База података је организована збирка подтака која описује стање неког система у реалном свету, било да је то број слободних хотелских соба, резултати биомедицинских истраживања, записи филмова на серверима и сл. Могу се поделити према садржају: библиографске, текстуалне, нумеричке и мултимедијске.

У ширем смислу, базе података се могу посматрати као интегрисани скуп података о неком систему и скуп поступака за њихово одржавање и коришћење, организован према потребама корисника. То је добро структурисана колекција података, која постоји једно одређено време, која се одржава и коју користи више корисника или програма.

Алати за уређивање, стварање и управљање базама података обједињени су под једним називом **СУБП - систем за управљање базама података** (енг. DBMS - *Database Management System*). Неки од најпознатијих СУБП-а су: Oracle, DB2, MySQL, Informix, PostgreSQL i SQL сервер.

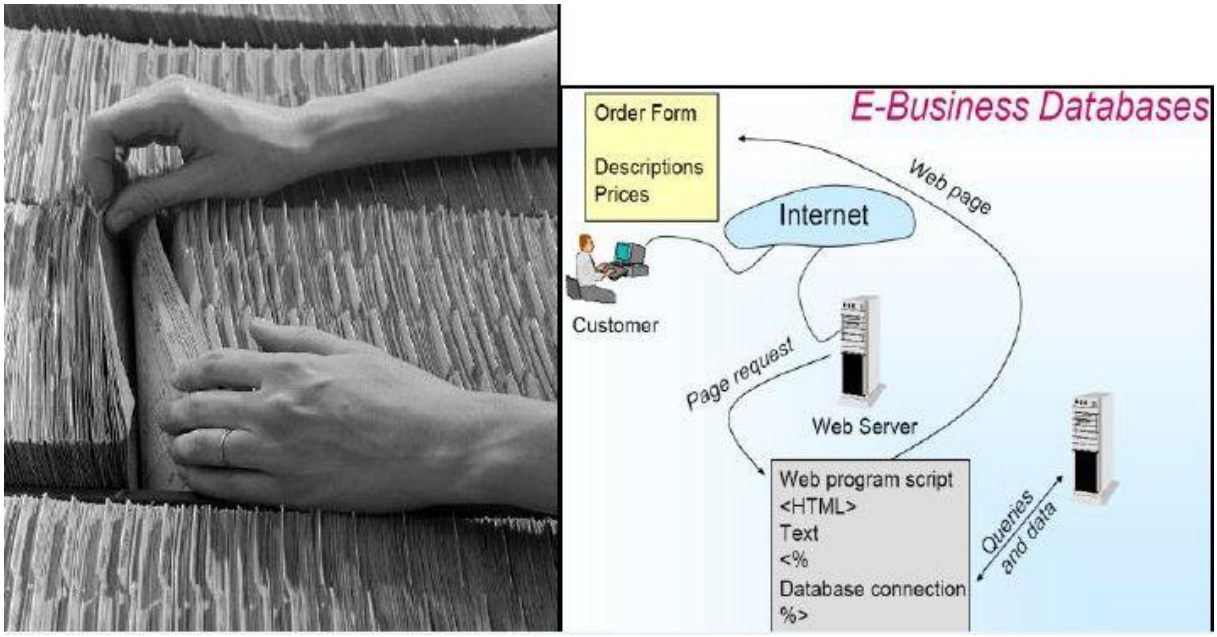
Честа појава од кад постоје базе података су напади злонамерних корисника који желе доћи до информација смештених унутар СУБП-а. Последице таквог напада могу довести до крађе идентитета, бројева кредитних картица, финансијских губитака, губитака приватности, нарушавања националне сигурности те других бројних опасних последица које су резултат приступа осетљивим подацима.

Како су се развијали СУБП-ови, тако је постајала све очитија потреба за њиховом сигурношћу. Због тога су сваком новом верзијом (код свих произвођача) додаване бројне сигурносне опције, као и сигурносне надоградње које би уклањале рањивости. Све већом употребом Интернета, приватно и на радном месту, јавио се императив осигуравања база података од приступа из спољног света. Не само заштитом мрежних ресурса на којима се СУБП налази, као што су постављање ватрозида и ажурирањем програмских пакета на серверима, него и заштитом самог СУБП-а постављањем дозвола приступа и сигурносних политика.

Ризици за губитак података:

- Отказивање хардвера,
- Отказивање софтвера,
- Случајно брисање података,
- Вирусна инфекција,
- Неовлашћен приступ.

Заштита база података је од кључне важности за сигурност било којег информационог система из разлога што је сама количина осетљивих информација смештена у тим базама велика и што о њој често зависи јако велики број људи.



Слика1. *Базе података некада(лево) и данас (десно)* [4]

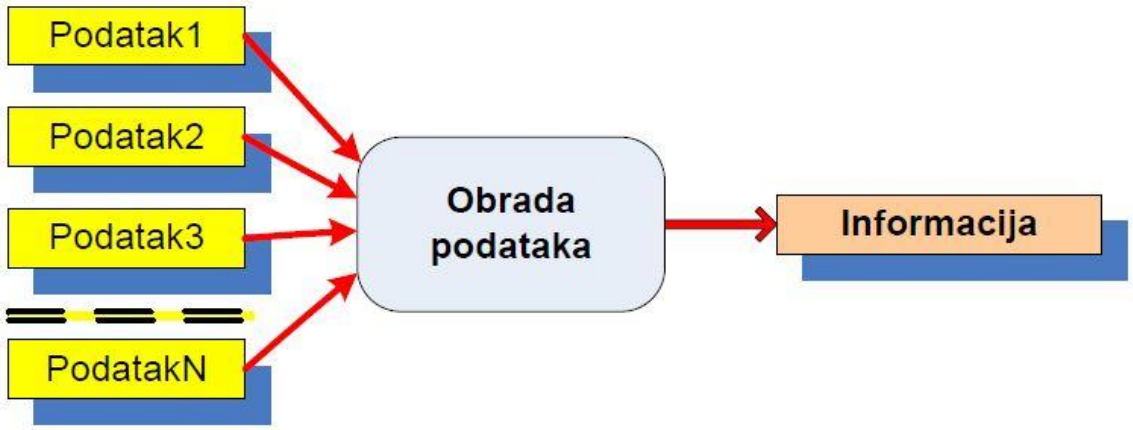
1. Базе података

1.1. Базе података-појам

Термини податак и информација су уско повезани и често се користе као синоними. Међутим, корисно је разликовати термине податак и информација.

Подаци су посебан скуп објективних чињеница о одређеном догађају и могу бити представљени у виду слова, бројева и симбола који самостално немају никакво значење. Подаци који су обрађени, категорисани и представљени у одређеном контексту и на одређен начин чине информацију [4] *(слика2).*

Информација је скуп података са одређеним значењем и представља поруку у било ком облику [4].



Слика2. *Обрадом података настаје информација* [4]

Базе података се користе за прикупљање, чување и манипулацију подацима на основу којих се добијају нове информације у различитим организацијама, као што су пословни системи, здравство, школство, владине институције итд. Свакодневно их користе појединци путем личних рачунара, радне групе путем мрежних сервера и сви запослени путем апликација које се налазе у пословним системима. Базама података такође приступају купци и други удаљени корисници коришћењем различитих технологија као што су говорни аутомати, web читачи, дигитални телефони и сл. Због велике конкуренције у свим областима пословања, може се очекивати да технологија база података добије још већи значај [8].

Све информације данас су сачуване у облику података у базама података.

За чување и обраду већих количина података користио се систем датотека (енг. *file*) код кога су се подаци чували и претраживали у скупу неповезаних датотека.

Основни недостаци овакве обраде података су:

- редунданса података односно вишеструко памћење истих података (нпр. подаци о радницима се налазе и у датотеци за обрачун зарада и у датотеци са распоредом радних места), што доводи до проблема у ажурирању (када се неки податак промени, то се мора учинити на свим местима на којима се он чува) али и повећању трошкова обраде података и утрошка меморијског простора;

- зависност програма од организације података тј. логичке и физичке структуре података (промене у структури датотеке тј. Убацивање новог поља у запис захтева и неминовну промену програма за обраду ове датотеке, иако нпр. он тај нови податак не користи);

- ниска продуктивност у развоју информационог система (уколико се неопходни подаци налазе у различитим датотекама, на разним медијумима, са различитим физичким организацијама задовољење и неког једноставног информационог захтева изискује значајне програмерске напоре);

- над структуром података представљеном великим бројем независних датотека веома је тешко развити софтверске алате за брз и ефикасан развој апликација и за непосредну комуникацију корисника са системом.

У рачунарству, базе података се деле на неколико врста :

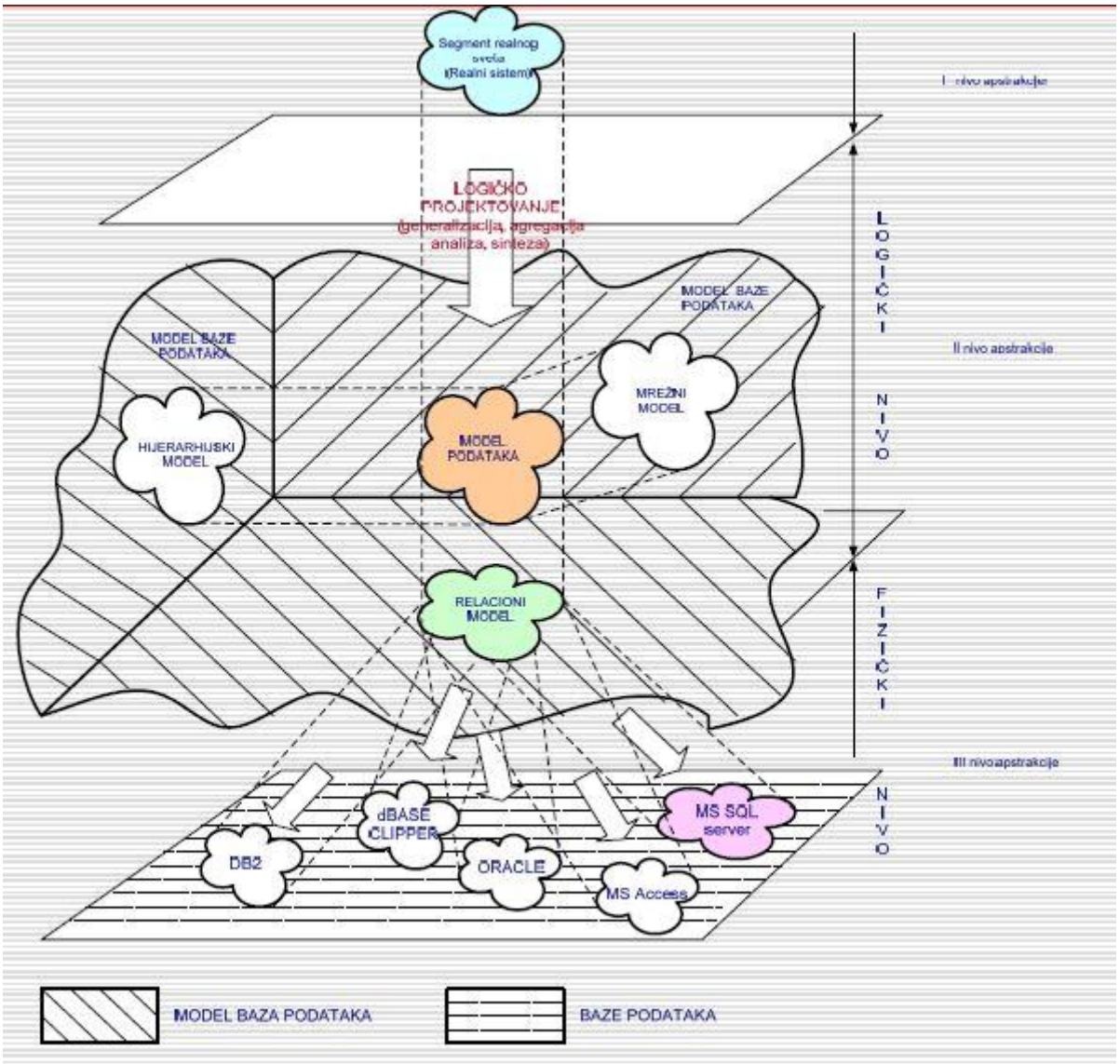
- **релацијско–табличне** - структуре које су дефинисане тако да се информацијама може помоћу логичких израза једноставно приступити и издвојити тражене информације,

- **објектно оријентисане** - користе се при програмирању, односно садржавању информације које припадају објектима, разредима и подразредима,

- **дистрибуиране** - базе података које се могу распоредити на више тачака у рачунарској мрежи или на Интернету.

1.2. Модели података

Информациони систем мора да представља адекватан модел реалног система у коме делује. Да би задовољио све захтеве који се пред њега постављају информациони систем мора бити аналоган модел реалног система. Од момента када се појавио до данас релациони модел база података заузео је најзначајније место и то као теоријска основа развоја и као конкретна практична реализација. Релациони модел база података је модел једноставног и прихватљивог концепта са јасно и свеобухватно дефинисаним формалним апаратом [3].



Слика3. *Апстракција као методолошки приступ у моделовању података [3]*

Систем база података се може посматрати као хијерархија апстракција. Сваки ниво у тој хијерархији апстракција је једна врста модела и представља скуп објеката (ентитета) и операција (радњи) над тим објектима као и веза између објеката [3] (што је приказано на *Слици 3.*).

За сваки ниво апстракције може се поставити питање могућности имплементације на нижем нивоу апстракције. Задатак пројектовања је да омогући рачунарску (физичку) реализацију најнижег нивоа апстракције, омогући трансформацију модела на вишем нивоу апстракције у модел на нижем нивоу апстракције. Један ниво те хијерархије апстракција је и модел података или логичка слика података. На нивоу непосредно испод модела података налази се репрезентација модела погодних структура података. Карактеристика другог нивоа је потпуно одсуство било каквих утицаја особина рачунарског система на модел података. На трећем нивоу су те особине присутне па се због тога тај ниво зове ниво физичке репрезентације. Физичка репрезентација се налази на најнижем нивоу хијерархије апстракција. Подаци су представљени у бинарном облику уз присуство свих специфичних карактеристика рачунарских меморијских медија и других компонената рачунарског система [3].

Новији модели података (модели треће генерације), који омогућавају боље моделирање објеката реалног света и веза између њих, су:

- Модел ентитети-везе (модел објекти-везе МОВ (енг.*ER-ENTITYRELATIONSHIP MODEL*) – најчешће се користи,
- SEMANTIC DATA MODEL (SDM),
- Проширени релациони модел,
- Објектно-орјентисани модел.

1.3.1.Реалциони модел

Релациони модел (енг. *Relation model*) представља скуп табела које описују систем тј. објекте и везе између њих. Иако је релативно једноставан и сиромашан, најчешће је коришћен и то пре свега јер се лако математички описује, операције над њим су једноставне захваљујући добро развијеном математичком апарату релационе алгебре и лако се имплементира на рачунару.

Пример релационог модела [2]:

P (P SIF, IME, BR NASLOVA, DRZAVA)
I (I SIF, NAZIV, STATUS, DRZAVA)
K (K SIF, NASLOV, OBLAST)
KP (K SIF, P SIF, R BROJ)
KI (K SIF, I SIF, IZDANJE, GODINA, TIRAZ)

Прве три релације представљају типове ентитета PISAC, IZDAVAC и KNJIGA редом, и имају сличну структуру као и одговарајући типови ентитета. Нов је први атрибут у свакој од релација, који представља шифру, тј. јединствени идентификатор писца, издавача и књиге. На пример, шифра књиге, K_SIF, може бити ISBN, P_SIF и I_SIF могу бити матични број писца и издавача. Релација KP представља апстрактни тип ентитета AUTOR, тј. однос између типова ентитета KNJIGA и PISAC, док релација KI представља апстрактни тип ентитета IZDAVASTVO, тј. однос између типова ентитета KNJIGA и IZDAVAC.

P

P_SIF	IME	BR_NASLOVA	DRZAVA
p1	B.Ćopić	2	Jugoslavija
p2	M.Benson	1	Engleska
p3	B.Šljivić-Šimšić	1	Jugoslavija
p4	D.Maksimović	2	Jugoslavija
p5	C.J.Date	1	Amerika

I

I_SIF	NAZIV	STATUS	DRZAVA
i1	Prosveta	30	Jugoslavija
i2	Addison Wesley Publ. Comp.	20	Amerika
i3	Dečje novine	10	Jugoslavija
i4	Matica srpska	30	Jugoslavija

K

K_SIF	NASLOV	OBLAST
k1	Osma ofanziva	roman
k2	Nemam više vremena	poezija
k3	Pionirska trilogija	roman
k4	Srpskohrvatsko-engleski rečnik	leksikografija
k5	An Introduction to Database Systems	računarstvo
k6	Tražim pomilovanje	poezija

KP

K_SIF	P_SIF	R_BROJ
k1	p1	1
k2	p4	1
k3	p1	1
k4	p2	1
k4	p3	2
k5	p5	1
k6	p4	1

KI

K_SIF	I_SIF	IZDANJE	GODINA	TIRAZ
k1	i1	2	1965	10000
k2	i1	2	1974	7000
k3	i1	1	1975	10000
k4	i1	2	1979	10000
k5	i2	4	1986	5000
k6	i3	1	1966	3000
k6	i4	3	1988	5000

Слика 4. *Пример релационог модела [2]*

1.2. Систем за управљање базом података (СУБП)

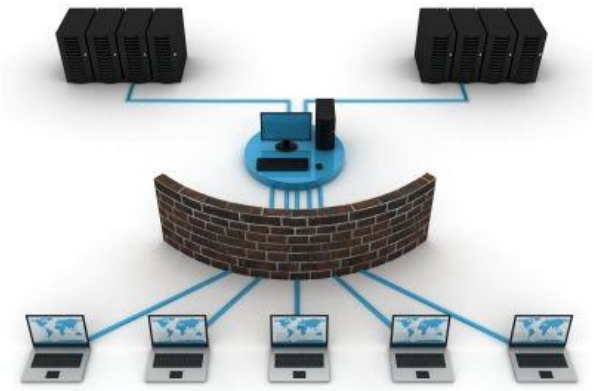
Систем за управљање базама података- **СУБП** (енг. DBMS - *Data Base Management System*) је софтверски систем који се користи за креирање, одржавање и манипулисање подацима, као и за контролу права приступа бази података [9].

- СУБП омогућава крајњим корисницима и програмерима да деле податке, тј. омогућава да се подаци користе од стране више апликација, а не да свака апликација има своју копију податка сачувану у посебним датотекама [9].

- СУБП такође пружа могућност контроле приступа подацима, осигурава интегритет података, успоставља контролу конкурентности и врши опоравак базе података. Програмери апликација за рад са базама података не морају да познају детаље о начину записа базе података на диску, не морају да формулишу алгоритме за ефикасан приступ подацима, нити су оптерећени било каквим аспектима око управљања подацима у бази података [9].

Данас је веома битан и значајан концепт базе података по коме је то, у ствари, заједнички ресурс кога истовремено (конкурентно) користи већи број програма, јер се прави ефекти базе података испољавају када се ради у мрежном окружењу. Ако се посматра база података једне банке у којој се налазе рачуни грађана, могуће је да се у истом тренутку на шалтеру у једној експозитури подиже новац са једног рачуна и уплаћује на други рачун, а да се истовремено у сасвим другој експозитури уплаћује новац на исти тај рачун. Поменути СУБП је управо ту да управља конкурентним радом више корисника и да обезбеђује синхронизацију њиховог рада.

У једноставнијим примерима обично је читав СУБП на серверском рачунару, док клијент коришћењем неког од упитних језика (најчешће SQL) врши упите и од сервера добија одговоре. У сложенијим системима, где се серверу истовремено обраћа веома велики број клијената па је тешко да на време свима одговори, примењују се вишеслојне апликације, код којих је главни сервер повезан са помоћним, апликационим серверима, који се старају о повезивању са базом, ауторизацији, трансакцијама, итд. и који се повезују са клијентским рачунарима и обрађују њихове захтеве.



Слика5.*Илустровани приказ базе подтака, СУБП-а, и корисника базе података [10].*

У раду сложених система неопходна је интеграција информација; ако се замисли рад једне велике компаније која се састоји из неколико фабрика; свака фабрика може имати различит СУБП, различите структуре података, чак и различите називе за исте ствари (нпр. у једној компанији за производњу хард дискова једна фабрика може брзину ротације својих производа изражавати у обртајима у секунди а друга уобртајима у минути). Централизовано управљање тј. прављење новог, јединственог система није увек решење, јер су можда фабрике до сада инвестирале значајна средства у развој базе података. Такође, могуће је, чак и када се уведе јединствени систем, да се компанији прикључи још нека нова фабрика са другачијим СУПБ-ом. Зато се најчешће, како би искористили податке из свих, већ постојећих база појединих јединица оне копирају у јединствену нову базу, звану складиште података (енг. *Data warehouse*), и то уз одговарајуће прилагођавање и усклађивање података (нпр. да се карактеристике производа изразе у истим јединицама за све фабрике).

Такође, СУБП има функцију да спречи штетне последице (нарушен интегритет базе, неконзистентно стање базе) при променама (трансакцијама) које се врше над базом података у вишекорисничком окружењу. У ту сврху постоје разне технике као што су техника закључавања података, техника временског означавања. Посебно је значајно управљање истовременим (конкурентним) трансакцијама. Тачност, заштита и доступност база података, као и коректност и перформансе трансакција које приступају тим базама су битни параметри за успех сваког пословног система.

Термини база података и управљање базом података се понекад мешају. Стручно говорећи, база података је увек скуп чињеница, а не рачунарски програм. СУБП је уведен као интерфејс између корисника (корисничких програма, апликација) и записа базе података на диску. Кориснички програми не приступају подацима директно, већ комуницирају са овим софтвером (програмом). СУБП управља структуром базе података: дефинише објекте базе, њихова својства (атрибуте), дозвољене вредности атрибута, везе између објеката, ограничења над објектима и међусобним везама. Омогућава манипулацију подацима у бази: уношење, брисање и измене, тј. омогућава њено одржавање. Контролише приступ подацима: ко може да приступи подацима, којим подацима и шта може са њима да ради. СУБП дозвољава дељење базе података између више апликација/корисника и чини управљање подацима успешнијим и делотворнијим. Уобичајено је да када се говори о софтверу за базе података, онда се мисли управо на СУБП. СУБП управља интеракцијом између крајњих корисника (апликација) и базе података. Крајњи корисници имају бољи приступ већем броју боље организованих података.

Да би СУБП превилно функционисао мора да:

- буде заснован на неком моделу,
- поседује језик за обезбеђење управљања БП,
- функционланост,
- адаптивност,
- перформантност,
- поузданост,
- погодност за одржавање ,
- постојаност речника,
- дуговечност.

1.4. SQL (Structured Query Language)-основне наредбе

SQL (енг. *Structured Query Language*) је стандардни релациони упитни језик. Настао је 1974.год у IBM-овој лабараторији и првобитно се звао SEQUEL (енг. *Structured English Query Language*). Крајем седамдесетих и почетком осамдесетих јављају се прве комерцијалне верзије релационих система, са SQL-ом као упитним језиком (нпр. Oracle, SQL/DS i DB2) [3].

- 1986.год. SQL је усвојен као стандардни релациони упитни језик(SQL-86),
- 1989.год. стандардизоване су и неке карактеристике језика које нису укључене првом стандардизацијом(SQL-89),
- 1992.год. следеца ревизија стандарда-после ове ревизије број стандарда је био 6 пута већи у односу на прву верзију (SQL-92),
- 1999.год. последња ревизија- укључени концепти објектне технологије, механизам тригера, рекурзија и процедурална проширења(SQL:1999) [3].

SQL је у сталном развоју, некада је био једноставан и близак кориснику и у великој мери декларативан (непроцедуралан) док се данас може рећи да је комплексан, процедурално/декларативан језик. SQL ради са табелама, које се могу креирати једном извршном наредбом.

Закључно са SQL-92 стандардом све наредбе SQL-а можемо сврстати у 3 категорије [3]:

1) **наредбе за дефинисање података,**
2) **наредбе за манипулисање (руковање) подацима,**
3) **наредбе за контролне (управљачке) функције.**

Наредбе за дефинисање података

Примери ових наредби су:

CREATE TABLE (креирање табеле базе података)
CREATE VIEW (креирање виртуалне табеле базе података-„погледа“)
CREATE INDEX (креирање индекса над комбинацијом колона табеле)
ALTER TABLE (измена дефиниције табеле)
DROP TABLE (избацивање табеле из базе података)

Наредбе за манипулисање подацима

Примери ових наредби су:

SELECT (приказ садржаја релационе базе података)
UPDATE (измена вредности колона табеле)
DELETE (избацивање редова колона табеле)
INSERT (додавање колона постојећој табели)

<u>Наредбе за контролне (управљачке) функције</u>

Примери ових наредби су:

GRANT (додела права коришћења сопствене табеле другим корисницима)
REVOKE (одузимање права коришћења сопствене табеле од других корисника)
COMMIT (пренос дејства трансакције на базу података)
ROLLBACK (поништавање дејства трансакције)

Последња група наредби (*Наредбе за контролне функције*) биће у наредном тексту објашњене јер су оне јако важне за тему овог рада.

2. Заштита података од уништења и оштећења

Базе података су критичне компоненте многих организација. Све од финансијских евиденција и важних информација клијената, до основних информација потребних за рад организације, чува се у базама података. Све више организација се ослањају на своје базе података да би могли да послују. Читаве компаније су банкротирале, затворале радњу, или су морале да отпусте радника због пропуста везаних за базу података, који је могао бити спречен да са сходно примењују план опоравка података. На базу података негативно може да утиче све од хардверских кварова, елементарних непогода, случајно или злонамерно брисње, до модификација. Корупција базе података дешава се када се то најмање очекује. Да би се заштитили подаци неке организације и обезбедили доступност базе података потребан је чврст и добро промишљен план за опоравак података [5].

Спашавање података је поступак опоравка података са оштећених, енкриптованих или на други начин недоступних секундарних медија за складиштење података. Најчешће се спашавање података врши са хард дискова, solid-state дискова, USB флеш диска, RAID система, трака, CD, DVD и осталих медија. Спашавање података се обично врши када је податке немогуће прочитати на матичном систему или другом рачунару. У пракси у највећем броју случајева подаци постану недоступни услед проблема на оперативном систему (ОС). Тада је процес спасавања података сведен на копирање података на други медиј. Нешто тежи, али свакако чест случај је да су подаци остали недоступни услед логичког проблема на партицији или фајл систему (логички кварови). Спасавање података се у случају логичког квара изводи на различите начине који имају за циљ формирање привременог стања у коме се реконструише дрво базирано на начину на који су подаци записани на диск (врста фајл система). Постоје случајеви када је потребно спасити обрисане податке. У највећем броју случајева обрисани подаци се заправо и даље налазе на медију, али су обрисани из дрвета или стабла фајл система. У случају да су подаци обрисани, а медиј је активно коришћен зависно од величине, врсте и једноставно места или организације фајл система одређене фајлове је могуће само парцијално спасити [11].

Најтежи проблеми су везани за губитак података на великим системима који укључују више засебних медија, најчешће су то серверски или enterprice хард дискови, везани у низ (RAID). Спасавање података са RAID система где су подаци остали недоступни услед природне катастрофе или људске грешке и немара су без премца најкомпликованији случајеви [11].

Неки од разлога због којих је неопходно заштити базу података [1]:

1. *Законски прописи, друштвни односи, и етичке норме налажу да се спречи неовлашћено коришћење многих података (* Законске мере заштите)

2. *Многи подаци у државној управи, предузећима и другим институцијама морају бити заштићењи (Управљање сигурносним ризицима)*

3. *У самом СУБП, неки подаци, на пример подаци у каталогу доступни су само администратору базе и неким специфичним апликацијама. Подаци о идентификацији корисника и њиховим лозинкама, такође морају бити заштићењи.*

2.1. Законске мере заштите

У Србији постоји релативно мали број закона и прописа који дефинишу информационе системе. Када су у питању базе података, први закон који је обухватио њихову заштиту објављен је 1998. године, где су први пут дефинисана и права произвођача базе података као предмет заштите сродних права (заснован на Директиви 96/9/ЕЦ Европског парламента из 1996.године о правној заштити базе података). 2012. године у оквиру **Закон о ауторском и сродним правима** [12] регулисане су посебне одредбе које се односе на базе података. Осим правне, постоје и техничке мере заштите базе података.

База података се дефинише као структурирана колекција података која постоји релативно дуго, коју користи и одржава више корисника односно програма и која се користи за складиштење информација. Разматрајући ову дефиницију, поставља се питање да ли је база података ауторско дело? Да би се нека збирка или колекција података прогласила ауторским делом или заштитила ауторским правом мора се утврдити њена оригиналност, без обзира уколико њени елементи нису заштићени ауторским правом. База података се састоји из делова тј. логично прикупљених целина које су на систематичан начин повезане и груписане у одређене катагорије у које је уложен интелектуални напор, креативност и лични допринос и које су доступне електронским или некм другим путем. Ове карактеристике су особине ауторског дела, па и заслужују да буду заштићене као интелектуална својина. Битно је напоменути да **Закон о ауторском и сродним правима** [12] под базом података не подразумева рачунарски програм који се користио за њено стварање или рад (члан 138, став 2).

Под произвођачем базе података се сматра физичко или правно лице које је сачинило базу података, уложило напор у сакупљање, проверу или презентацију њеног садржаја и података (члан 137.). Сваки произвођач базе података има имовинска права тако да другима може да забрани или дозволи умножавање, закуп или јавно саопштавање како целе базе података тако и неких њених делова, према члану 140. **Закона о ауторском и сродним правима** [12]. Сва права произвођача трају 15 година од настанка базе података. Уколико дође до додавања нових садржаја у базу података, брисања непотребих или промене појединих делова или целе базе добија се нова верзија базе података. Свака нова верзија доноси произвођачу додатних 15 година права на коришћење (члан 147, став 2 и 3). Уколико је база података објављена јавности, рок од 15 година задржавања права важи од момента када је објављена, а не од момента када је сачињена.

Закон о ауторском и сродним правима [12] Републике Србије не садржи специфична ограничења права произвођача базе података, мада и сам произвођач може прекршити одређене законске одредбе уколико се не наведе извор из кога су делови базе података преузети.

Објављивање садржаја базе података или неких њених делова, од стране корисника, могуће је само у случају поседовања одговарајуће лиценце или сагласности носиоца права. Јасно су дефинисана права и обавезе носиоца права и корисника базе података :

- Носилац права кориснику предаје базу података и потребну документацију која садржи упуства за унос, коришћење и претрагу података.
- Корисник може да направи копије базе података само по уговору и по договореном броју.
- Носилац права је дужан да замени базу података у случају оштећења, само ако корисник није одговоран за оштећење.

У пракси се често дешава да се поједини делови базе података објављују на интернету, у комерцијалне сврхе, ради рекламирања, обављања одређених делатности, без дозволе преради, објави или на било који начин искористи туђа база и сл. Ако су права произвођача базе података повређена, произвођач базе података има сва законска овлашћења да писменим путем затражи од корисника да не користи преузете материјале или да поднесе кривичну пријаву. У **Закону о ауторском и сродним правима** [12] Републике Србије предвиђене су санкције за привредни преступ због повреде права произвођача базе података и новчане казне (за правна лица од 100. 000 до 3.000. 000 динара; за предузетнике од 50.000 до 500.000 динара (члан 215. и 216.))

2.1.1. Правилник о чувању, заштити и сигурности података у оквиру информационог система Централног регистра обавезног социјалног осигурања

(Сл. гласник РС бр. 29/13) [11]
Основни текст на снази од 06/04/2013 , у примени од 06/04/2013

Садржина правилника

Члан 1.

Овим правилником уређују се процедуре чувања, заштите и сигурности података информационог система Централног регистра обавезног социјалног осигурања (у даљем тексту: информациони систем Централног регистра).

Циљеви заштите информационог система

Члан 2.

(1) Циљеви заштите информационог система Централног регистра су:

1) очување поверљивости података, чиме се онемогућава неауторизован увид и коришћење података из информационог система Централног регистра;
2) заштита интегритета података, чиме се онемогућава измена података и гарантује аутентичност података;
3) очување расположивости података, чиме се омогућава реконструкција података у случају њиховог намерног или ненамерног оштећења.
(2) Заштита из става 1. овог члана обезбеђује се кроз заштиту приступа рачунарској опреми и мрежи, која се реализује на мрежном нивоу кроз специјализоване хардверске компоненте и уз употребу протокола заштите, као и кроз заштиту приступа подацима.

Чување, заштита и сигурност података Централног регистра

Члан 3.

Централни регистар је одговоран за чување, заштиту и сигурност података у оквиру информационог система Централног регистра, што подразумева:

1) заштиту од неовлашћеног приступа ресурсима који су предмет заштите, њихово неовлашћено коришћење или
манипулације базом података информационог система од стране интерних и екстерних корисника;

2) заштиту интегритета података, њихову расположивост и неовлашћени увид у поверљиве податке;
3) заштиту базе података од вируса и осталих облика малициозних кодова;
4) осигурање преноса података из Јединствене базе интерним и екстерним корисницима;
5) чување података и управљање сигурносним копијама базе података у оквиру информационог система;
6) политику преносних рачунара у погледу приступа бази података Јединствене базе и чувања података у јединственој бази;
7) осигурање континуитета активности у случају пожара, поплаве, земљотреса или друге непогоде која се сматра резултатом више силе и која доводи до неуобичајеног прекида у раду информационог система;
8) повраћај сачуваних података у случају губитка, оштећења или уништења рачунарске опреме информационог система;
9) тестирање јединствене базе података ради откривања сигурносних проблема на редовној основи и након инсталирања нових верзија Јединствене базе података;
10) инсталирање софтверске надоградње ради уклањања сигурносних проблема који се установе на Јединственој бази у оквиру информационог система или на повезаном софтверу;
11) праћење сигурносних инцидената у бази података информационог система ради предузимања корективних мера;
12) управљање сигурносним инцидентима, едукација и обука свих овлашћених особа ради стицања потребних знања о чувању и сигурности података;
13) физички приступ и заштита базе података у оквиру информационог система и рачунарске опреме;
14) одржавање рачунарске опреме информационог система.

Мере заштите приступа информационом систему

Члан 4.

(1) Мере заштите приступа информационом систему Централног регистра су:
1) аутентификација - која представља процес утврђивања идентитета особе која жели да приступи информационом
систему Централног регистра;
2) ауторизацију - која представља право приступа и дозвољених операција за аутентификовано лице;
3) заштита тајности - што подразумева шифровање података у циљу спречавања неовлашћеног увида;
4) непорицање одговорности - што подразумева обезбеђење доказа да је неко извршио одређену радњу, односно трансакцију.

(2) Реализација система заштите информационог система Централног регистра подразумева обавезну примену квалификованих електронских сертификата за приступ преко Портала и аутентификацију трансакција, као и за аутентификацију приступа Веб сервисима.
(3) Аутентификација приступа сервисима од стране државних органа и организација, са којима Централни регистар врши размену података подразумева и обавезну примену серверских сертификата.

(4) Изузетно, осигураници и осигурана лица, који имају право увида у личне податке који се односе на осигурање, могу приступити информационим систему Централног регистра на основу јединственог броја и лозинке, које додељује Централни регистар.

Приступ информационом систему Централног регистра

Члан 5.

(1) Централни регистар управља корисничким налозима, правима приступа и корисничким лозинкама за интерне и екстерне кориснике јединствене базе Централног регистра.
(2) Централни регистар дужан је да обезбеди приступ подацима у оквиру информационог система само од стране
овлашћених лица.
(3) Сваки приступ информационом систему мора бити аутоматски забележен јединственим идентификатором лица у бази података јединственог система, са тачним временом приступа.
(4) О сазнањима у вези са покушајима неовлашћеног приступа информационом систему Централног регистра, администратори Јединствене базе података дужни су да обавесте овлашћено лице.

Физичка заштита података информационог система и чување безбедносних копија

Члан 6.

(1) Ради обезбеђења непрекидног функционисања информационог система, Централни регистар обезбеђује физичку заштиту података примарног информационог система, формирањем секундарне базе података и секундарног рачунарског система.
(2) Секундарна база података и секундарни рачунарски систем морају бити удаљени од места на коме се налази примарни информациони систем.
(3) Локације из ст. 1. и 2. овог члана морају бити на адекватан начин заштићене од пожара и поплава, као и имати 24-сатни безбедносни систем.
(4) Приступ локацијама на којима се налазе базе података информационог система и чувају безбедносне копије имају само овлашћена лица.

Одржавање, поправка и повлачење из употребе опреме за информациони систем

Члан 7.

(1) Централни регистар обезбеђује одржавање рачунарске опреме за информациони систем а, у случају поправки, претходно спрема безбедносне копије података, како би се спречио губитак података.
(2) Одржавање и поправка рачунарске опреме, врши се искључиво под надзором овлашћених запослених у Централном регистру.
(3) У случају повлачења рачунарске опреме из информационог система, сви подаци претходно морају бити трајно и сигурно избрисани.

2.2. Управљање сигурносним ризицима

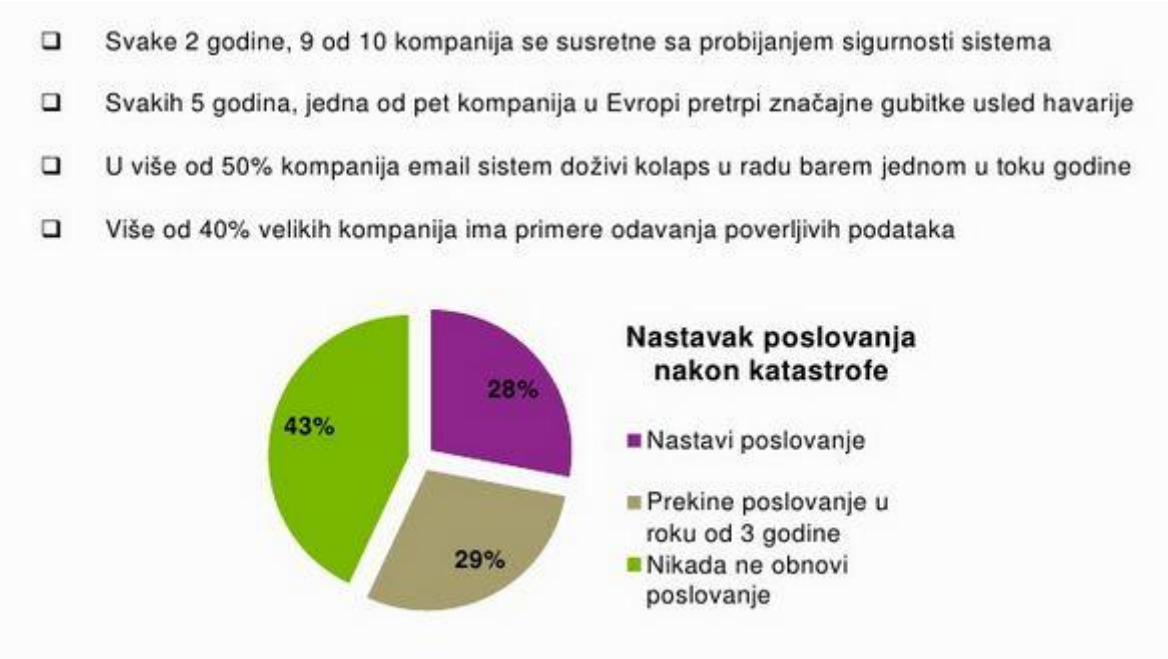
Процена IT конторола и управљањем ризицима информационих система[14]

IT ризику највише су изложене организације које се у свом пословању у великој мери ослањају на информационе системе и имају потребу да ИТ сервиси буду доступни у континуитету, као и **обрадом масовних трансакција** на дневном нивоу.

Успостављање адекватних ИТ контрола треба да буде одговор организације на ИТ ризик.

Примери ИТ ризика:

- ослањање на системе и програме који нетачно обрађују податке
- неауторизован приступ подацима
- кршење правила поделе дужности
- пропуст да се направе неопходне измене у системима или програмима
- потенцијалан губитак или немогућност приступа подацима, софтверу или хардверу
- недоступност потребних људских ресурса



Слика6.*Проблеми пословања због лоше безбедности података [14]*

Независна контрола ИТ операција као и процена ИТ сервиса и степана њихове подршке процесима организације су витални за заштиту ИТ ресурса, информација, ИТ система као и побољшање пословања.

Управљање ризиком треба да обухвати целокупни информациони систем организације и све ИТ ресурсе. То је континуирани процес упоређивања процењених ризика с предностима и трошковима предложених мера у складу с пословним циљевима.

Овај процес генерално обухвата:

- процену ризика,
- смањивање ризика,
- одржавање прихватљивог нивоа ризика.

Потребно размотрити различите врсте заштитних мера, спровести анализу и узети у обзир прихватљив ниво резидуалних ризика.

Заштитне мере се бирају зависно од вероватноће појаве нежељеног догађаја и од његовог утицаја на информациони систем и нарушаавање његове функционалности и сигурности.

Слика7. *Предлог приступа [14]*

2.3. Заштита на нивоу СУБП

Као што је већ споменуто СУБП пружа могућност контроле приступа подацима, осигурава интегритет података, успоставља контролу конкурентности и врши опоравак базе података.

У наредном одељку биће објашњени основни концепти сигурности база података, а детаљније биће приказани најзначајнији механизам сигурности.

Основни концепти сигурности база података [1]:

- Сигурност целокупног система
- Модел сигурности базе података.
- Дискрециони механизам сигурности (Discretionary mechanism)
- Механизам овлашћења (Mandatory mechanism)
- Статистичке базе података
- Енкрипција података.
- Регистар и ревизија корисникових акција (Audit Trail).

2.3.1. Сигурност целокупног система

Један од проблема сигурности је и заштита целокупног система, односно спречавање неовлашћеној особи да приступи систему било са циљем да добије неке информације или да оштети део или целу базу података. Сигурносни механизам СУБП мора да спречи овакав приступ. То се остварује додељивањем корисничких идентификација и њима одговарајућих лозинки. Проблем сигурности такође обухвата и физичку заштиту система, односно спречавање физичког приступа неовлашћењним особама месту са опремом на којој се чува одговарајућа база [1].

2.3.2. Модел сигурности базе података

Оснвни модел сигурности базе података се може дефинисати преко скупа ауторизација односно скупа четворки,

Ауторизација:<корисник, објекат, операција, услов>

Ауторизација исказује којим објектима базе посматрани корисник може да приступи, које оперције над њима може да изврши и под коим условима. Поред основног постоје и други модели сигурности, који ће укратко бити описани. Део СУБП који обавља овај задатак назива се подсистем сигурности (енг. *security subsystem*) или подсистем ауторизације (енг.*autorization subsystem*) [1].

2.3.3. Дискрециони механизам сигурности (Discretionary mechanism)

Дискрециони механизам сигурности подржава основни модел у коме се појединим корисницима дају специфична права приступа за различите објекте [1].

SQL standard подржава само дискрециони механизам сигурности преко наредбе GRANT за креирање ауторизације и наредбе REVOKE којом се ауторизација укида.

Синатакса GRANT наредбе је:
GRANT< lista privilegija>
ON<objekat baze>
TO<lista identifikatora korisnika>
[WITH GRANT OPTION];

Објашњења:

- Привилегије могу да буду: USAGE, SELECT, UPDATE, DELETE, REFERENCES, ALL PRIVILEGES. Привилегија USAGE се даје за коришћење неког специфичног, корисничког (дефинисаног преко SQL-а) домена. Привилегија REFERENCES дозвољава реферисање неке табеле при дефинисању правила интегритета, на пример референцијалног интегритета. Остале привилегије су очигледне.
- Објекти базе могу да буду DOMAIN <ime domena> и TABLE <ime tabele>. Под табелом се не подразумева само базна табела већ и поглед.
- У листи идентификатора корисника наводе се сви корисници којима је дозвољен исти скуп привилегија над истим објектима. Листа идентификатора може бити замењена са кључном речи PUBLIC, чиме се дефинисане привилегија над листом објекта дају свим корисницима система.
- WITH GRANT OPTION исказ дефинише могућност да корисници из листе корисника преносу ауторизацију на неке друге кориснике, односно да сами издају GRANT наредбу.

Ако се упореди синтакса SQL-а са претходно датим општим моделом сигурности базе података, види се да SQL не подржава постављање услова ауторизацији. Тај недостатак се за табеле може решити претходним дефинисањем погледа са датим условом, над којим се примењује GRANT исказ.

Привилегија дата неком кориснику се ноже опозвати коришћењем наредбе REVOKE који има следећу синтаксу:

REVOKE [GRANT OPTION FOR] <lista privilegija>
ON <objekat baze>
FROM <lista identifikatora korisnika>
<opcija>;

Објашњења:

- Опција GRANT REVOKE се користи када се опозове могућност преноса провилегија
- <opcija> може да има вредност RESTRICT или CASCADES и дефинише начин на који се поступа са пренетим привилегијама за привилегије које се опозивају. Опција RESTRICT онемогућава опозив привилегије ако је она пренетаа другим корисницима. При опцији CASCADES опозивају се, поред посматраних, и све пренете привилегије.
- Остали делови наредбе REVOKE имају исто знчење као и у наредби GRANT.

Власник шеме базе података (корисник који је креирао шему) има све привилегије у њој. Такође може и да извршава све операције над табелама и погледима у шеми. Власник преноси привилегије на кориснике, а ови могу тај ланац давања привилегија да наставе, ако су за то овлашћењи.

Пример за процес давања и опозивања привилегија:

Нека је Milos власник шеме StudentskiIS која има следеће табеле:

Student (BI, Ime, Starost, Semestar, Smer)
Predmet (SPred, NayivPred, BrojCas)

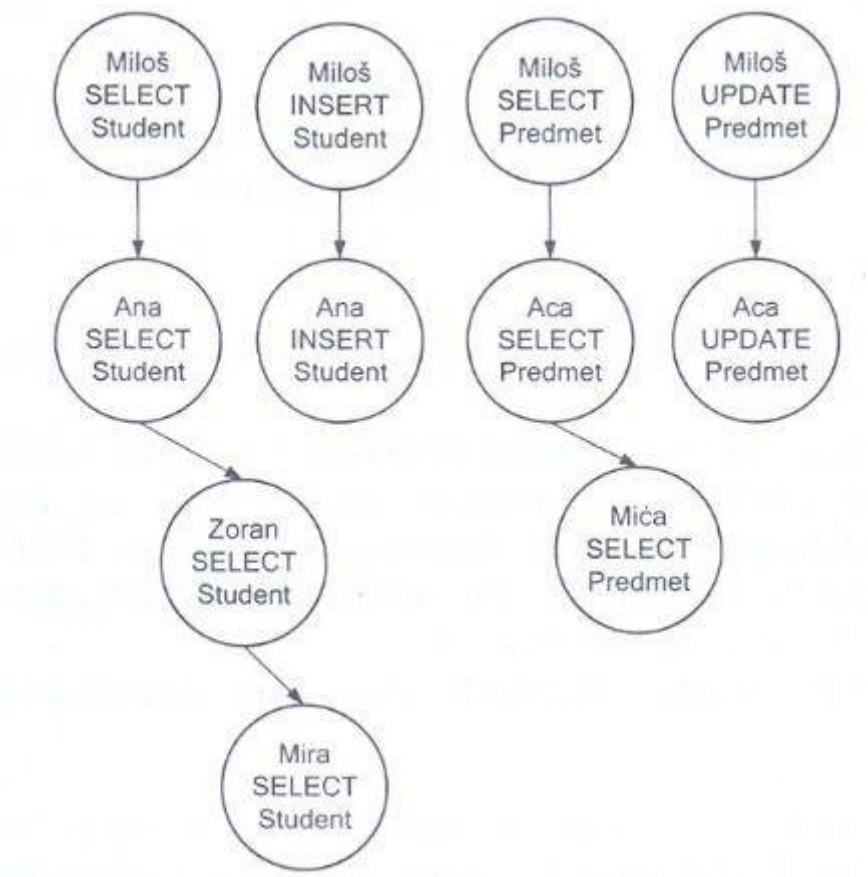
Milos даје привилегију SELECT и INSERT Ani за табелу Student, а Aci привилегије SELECT и UPDATE за табелу Predmet, дозвољавајући им да те привилегије пренесу и на друге кориснике

```
GRANT SELECT, INSERT ON Student TO Ana
        WITH GRANT OPTION;
GRANT SELECT, UPDATE ON Predmet TO Aca;
        WITH GRANT OPTION;
```

Ana преноси привилегију SELECT за табеле Student на Zorana са опцијом преноса на друге, а овај преноси привилегију SELECT за табелу Student на Miru. Aca преноси привилегију SELECT за табелу Predmet на Micu.

```
GRANT SELECT  Student TO Zoran
        WITH GRANT OPTION;
GRANT SELECT Student TO Mira;
GRANT SELECT Predmet TO Mica;
```

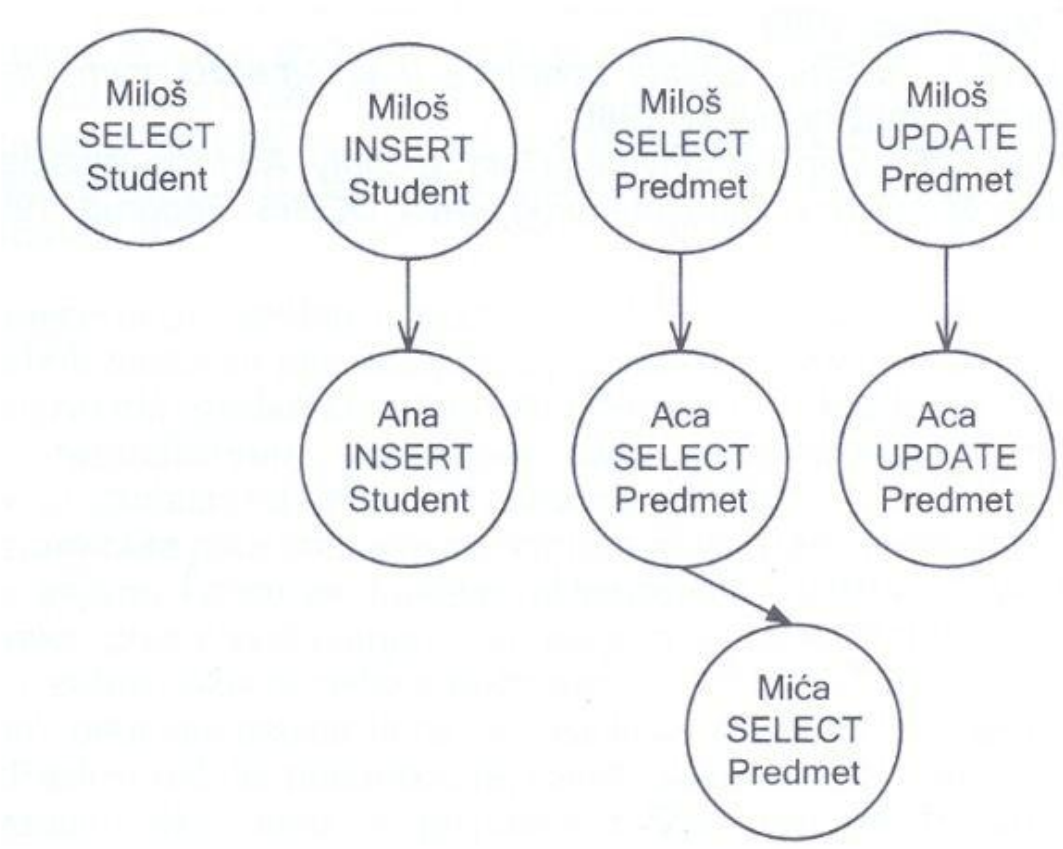
Давање и пренос привилегија биће приказани преко „GRANT" дијаграма (графа) на *Слици 8.* Чвор овога графа представља једну привилегију корисника, а усмерена грана приказује пренос привилегија на другог корисника.



Слика8. *Grant дијаграм за приказани пример [1].*

На Слици 9. приказан је GRANT дијаграм за исти пример, после повлачења привилегија Zoranu и Ani за SELECT Student и Aci за SELECT Predmet преко наредби

```
REVOKE GRANT SELECT ON Student
        FROM Ana
         CASCADES;
REVOKE GRANT SELECT ON Predmet
      FROM Aca
      RESTRICT;
```



Слика 9. *Grant дијаграм после повлачења привилегија [1].*

Као што се са *Слике 9.* види, опција CASCADES укида и све привилегије које је Ана пренела, док опција RESTRICT не дозвољава укидање привилегије Аци за SELECT ON Predmet због тога што је он ту привилегију пренео Мићи.

2.3.4. Механизам овлашћења (Mandatory mechanism)

Механизам овлашћења (енг. mandatory mechanism) подржава специфичан модел сигурности у коме је, с једне стране сваком објекту базе података додељен неки степен „тајности“ (на пример државна тајна, строго поверљиво, поверљиво, јавно), а са друге корисницима дато право да приступе објектима дефинисаног степена тајности. Подразумева се „хијерархија овлашћења“, корисник са овлашћењем вишег степена, може да приступи и свим објектима нижег степена тајности [1].

2.3.5. Статистичке базе података

Статистичке базе података се, са тачке гледишта сигурности, посебно анализирају. Статистичке базе података дају статистичке, збирне информације о феноменима за групу ентитета дефинисане по неком критеријуму (популацији). На пример, нека статистика становништва даје неке збирне податке о популацијама становништва, дефинисане на основу старости, прихода, нивоа образовања и слично. Механизам сигурности статистичких база мора да онемогући да се из збирних података извуку закључци о вредностима појединачних података за неки ентитет. То се најчешће постиже спречавањем добијања статистичких информација за популацију мању однеког унапред задатог броја и/или спречавањем извршеа секвенце упита над истом популацијом. Проблем сигурности статистичких база података добијаће на значају са већим коришћењем „стоваришта података“ (енг. data warehouse) [1].

2.3.6. Енкрипција података- шифровање података

Да би се осигурали посебно осетљиви подаци у бази података примењује се енкрипција података, специфичан начин кодирања података, који је тешко „разбити“ [1].

У системима база података, осетљиве информације се чувају нешифроване и може бити рањив на нападе. Без озбзира величина предузетих мера, увек ће бити безбедносне пукотине које нападачи могу да користе да продре у базу података. Да би избегли ризик од ових претњи препоручује енкрипцију података.

Многи сигурносни механизми спречавају приступ и имају компликоване процедуре које омогућавају приступ само овлашћеним корисницима.Криптографија дозвољава да неовлашћена особа има пун приступ поруци, али и даље сигуран систем.

2.3.6.1. Основи криптографије

Криптографија је научна дисциплина која проучава методе очувања поверљивости. Сама реч криптографија је грчког порекла и потиче од речи κρυπτω cripto(што значи тајно или скривено) и γραφω graf (што значи писати) или λεγειν legein (значи говорити). То је мултидисциплинарна област, јер у себи садржи многе научнотехничке дисциплине, а посебно у вези са математиком, и недавно за информатику и рачунарство. Модерне криптографске методе се односе на криптографские алгоритаме. Сваки криптографски алгоритам садржи мало података, трансформацију која се зове кодирање и декодирање. Кодирање (шифровање) је процедура која трансформише оригиналну информацију (отворени текст енг. plaintext) у шифроване податке (шифрат енг. ciphertext) или криптограм [7].

Обрнути процес, дешифровање (дешифровање), реконструише отворени текст на основу шифрата. Када је енкрипција, поред отвореног текста, користи другу независну вредност која се зове кључ шифровања. Слично томе, трансформација дешифровати, користећи кључ за дешифровање. Број симбола који представљају кључ (дужина кључа) зависи од криптографског алгоритма и параметара система безбедности. Криптографских алгоритама су обично доступни јавности и дешифровање зависи од тајног кључа, што указује да поред заштите отвореног текста и кључ треба да буде заштићен. Али, као у старим данима, ни данас не постоји потпуно сигуран алгоритам за незаконито добијање поруке преко шифрата. Ипак, из доброг алгоритма захтева да цена за разбијање шифри прелази вредност информација за које ради, и да је време за ту акцију требало да буде довољно дуго тако да добијена информација постане бескорисна и застарела. Безбедносне енкриптованих података не зависи само од алгоритма који се користи, већ и од тога како се спроводи.

Постоје три опште категорије криптографских алгоритама који се користе у системима база података, и то: симетрични, асиметрични и *hashing* алгоритми. Сваки криптографски алгоритам има своје предности и своје мане. Неки системи користе комбинацију ових алгоритама и тако искористе своје предности. Такви криптографски системи се називају хибридни системи [7].

2.3.6.2. Симетрични алгоритми

Симетрични алгоритми користе исти кључ за шифровање и дешифровање података. Еквивалент имена за симетричне алгоритаме су: *conventional*, *secret key*, *classical* и *private key*. Као што име сугерише, код ових алгоритама алгоритма кључеви мора да буду скривени или тајни. Ако се деси да до кључа дођу стране који нису укључене у комуникацију, сигурност података се смањује или потпуно елиминише. Другим речима, свако ко поседује овај дељени тајни кључ има способност да чита (и пише) шифроване поруке. Стране које учествују у комуникацији и које деле тајни кључ се обавезује на заштиту приступа кључу. Код ове врсте алгоритама поставља се проблем како пренети тајни кључ. Проблем је у томе ако се тајни кључ пресретне спорука може да се прочита. Због тога, овај тип енкрипције најчешће се користи у заштити података који се не дели са другима. Предност ових алгоритама лежи у њиховој брзини, а користе се код великих фреквенција читања и писања у базу података [7].

Дужина кључа је изузетно важна за безбедност података. Постоје два основна начина за напад података шифровани симетричним алгоритмом. Први тип напада је напад „нагађање и провера“, а обично је успешна када алгоритам спроводи се на основу нетачних и лоше произвољно изабраном кључу. Овде нападач нагађа који је кључ коришћен за шифровање и затим покушава да примени кључ и провери да ли су ти кључеви коришћени. Напад са бруталном силом (енг. bruto force attack) је друга врста напада на криптографски систем и заснива се на проверавању свих могућих кључева.

Ови напади су постали успешни након повећане снаге рачунара, што је омогућило стварање и проверавање броја кључева у релативно кратком временском периоду. Ако се користе базе података са симетричним алгоритмом, препоручује се минимална дужина кључа од 128 бита. Све мање ставља податке у непотребан ризик [7].

2.3.6.3. Асиметрични алгоритми

За разлику од алгоритама са тајним кључем који користе један дељени кључ, асиметрични алгоритми користе два кључа. Један од кључева је јавни, други приватни. Ови алгоритми се зову алгоритми са јавним кључем, а њихов принцип рада је следећи: на основу тајног кључа који поставља прималац, генерише се јавни кључ. Јавни кључ се даје старни која шаље шифроване податке примаоцу. Помоћу њега, пошиљалац шифрира информацију коју жели да пошаље и такву шаље примаоцу. Кад шифрат стигне примаоцу он га дешифрује помоћу свог тајног кључа. Значи, тајни кључ има само прималац, а јавни кључ може имати било ко, пошто се он користи само за шифровање, а не и за дешифровање. Предност овог начина шифровања је у томе што не постоји брига у случају да неко пресретне јавни кључ, јер помоћу њега може само да шифрује податке. Асиметричне операције су знатно спорије него операције код симетричних алгоритама, што их чини не погодним за стратегије енкрипције база података, зато што могу значајно утицати на перформансе базе података. Системи са јавним кључем су се показали као спори уколико се поља често дешифрују у реалном времену. Ако се подаци пишу једном и онда ретко користе, тада се као опција могу користити асиметрични алгоритми [7].

2.3.6.4. Криптографске функције за сажимање

Криптографске функције за сажимање – хеш (енг. hash) функције се сврставају у криптографске алгоритме без кључа. Hash function означава функцију која компресује низ података произвољне дужине у низ података фиксне дужине. Хеш функције се користе код заштите интегритета података и раде по следећем принципу; када нови податак пристигне он се хешира и онда се се пореди са хеш вредностима оригинала. Ако подаци нису корумпирани или измењени вредности ће бити идентичне, а ако су подаци били измењени, хеш вредности биће различите. Пример кориштења *hashing* је провера password -а између радне станице и сервера. Коришћењем ове технике информација никада не „путује“ кроз мрежу у облику отвореног текста, па нема опасности од пресретања. Како *hashing* алгоритми не користе кључ тако ни дужина кључа није проблем, а пошто *hashing* није реверзибилан процес нема ни опасности од дешифровања. *Hashing* функције имају мало оптерећење код израчунавања: утицај израчунавања хеш-ева унутар система за управљање базама података (СУБП) може се грубо упоредити са симетричном енкрипцијом, али без постојања проблема управљања кључевима. Величина вредности излаза *hashing* процеса је фиксна [7].

2.3.6.2. Метадологија енкрипције

Обзиром на податке који се криптују и на нивоу на којем се криптовање обавља, постоје различите врсте енкрипције које су применљиве у заштити система база података, а у складу са тим и следеће поделе [7]:

- енкрипција за заштиту података током преноса (енг. data–in–motion));
- енкрипција за заштиту података током мировања (енг. data–at–rest);
- енкрипција на нивоу програмског интерфејса, тј. енкрипција на нивоу апликације (енг. aplication – layer encryption);
- енкрипција на нивоу базе података (енг. database–layer–encryption);
- енкрипција фајла (енг. . file–based encryption);
- транспарентна енкрипција.

У зависности од алата који се користи енкрипција се може поделити и на:

- енкрипција на нивоу софтвера;
- енкрипција на нивоу хардвера.

Софтверска енкрипција се у области информатике користи да означи процес шифровања података у самом процесору, а ако се подаци шифрују помоћу хардвера прикључених на рачунар онда се ради о хардверској енкрипцији. Иако је хардверска енкрипција бржа у односу на софтверску енкрипцију, она може бити недоступна, па се стога користи софтверска енкрипција.

2.3.6.3. Заштита података током преноса

Енкрипција података током преноса штити податке током преноса од базе података до клијента и обрнуто. Пренос података се може обављати нпр. кроз локалну мрежу, Интернет или преко бежичних мрежа. Енкрипција података током преноса неопходна је да се спречи сазнавање садржаја поруке након пресретања саобраћаја, када се подаци преносе од клијента до базе података и обратно. Овај вид енкрипције се највише користи у заштити података пошто се не веже за апликације и захтева минималне напоре за развијање. Постоје различити сигурносни протоколи за ову врсту енкрипције, као што су: SSL (eng. Secure Sockets Layer), TLS (eng. Transport Layer Security) и IPSEC (eng. Secure Internet Protocol) [7] .

2.3.6.4. Заштита података током мировања

Термин data–at–rest се користи да означи све податке који се налазе унутар система база података, а искључује податке који се преносе кроз мрежу или се привремено чувају у меморији док чекају на операције читања или ажурирања. Према томе, енкрипција података који мирују је енкрипција информација које се чувају у системима база података.

Постоје три најчешће прихваћене стратегије енкрипције података који мирују, и то [7]:

- енкрипција на нивоу апликације;
- енкрипција на нивоу базе података;
- енкрипција на нивоу *storage*, односно на нивоу фајла.

2.3.6.5. Шифровање на нивоу апликације

Енкрипција на нивоу апликације омогућава да се подаци енкриптују и као такви смештају у базе података. Слично, апликација враћа енкриптоване податке из базе података и онда их декриптује. Енкрипција на нивоу апликације може захтевати преправку постојећих апликација ако у њих нису уграђене енкрипционе/декрипционе способности или куповину апликација које поседују способности енкрипције/декрипције. Захтев преправке постојећих апликација је непрактичан услед ограничених ИТ ресурса, недостатка приступа изворном коду или недостатка познавања старог кода. Преправљање апликација је такође скупо, ризично и уноси фактор кашњења имплементације. Ако су подаци енкриптовани на нивоу апликације, онда све апликације које приступају енкриптованим подацима морају бити промењене да подрже модел енкрипције/декрипције. Према томе, јасно је да је током фазе планирања примене енкрипције, потребно одредити које апликације имају потребу приступа енкриптованим подацима, затим је потребно обезбедити ресурсе да се модификује целокупни скуп апликација од којих се захтева да пруже способност енкрипције на нивоу апликације. Ако се примени исправно, енкрипција на нивоу апликације штити податке од: напада на *storage*, крађе *storage media* и напада од злонамерних администратора база података. Овај приступ је потпуно транспарентан у односу на базе података, а то значи да није потребно ништа радити на нивоу базе података осим обезбиедити да су поља (величина поља) способна да приме шифрован податак (који је обично већи него еквивалентан отворени текст). Ипак, овај приступ има и недостатке као што је рецимо могућност коришћења тих података само кроз апликацију, а то значи да се не могу користити SQL едитори или неки алати администратора база података. Међутим, ово се може посматрати и као предност, пошто се спречава нпр. напад од стране злонамерних администратора. Недостатак енкрипције на нивоу апликације огледа се и у томе да, пошто се декрипција обавља на страни клијента, хакери који компромитују клијентске апликације имају приступ дешифрованим подацима. Да би се сприечио овај недостатак потребна је примена снажне аутентификације на апликационом нивоу [7].

2.3.6.6. Шифровање на нивоу базе података

Енкрипција на нивоу базе података омогућава да се осигурају подаци док се они уписују, односно читају из базе података. Остварује се позивањем функција, ускладиштених процедура (енг. stored procedure) и окидача (енг trigger) базе података, које обично развијају продавци „трећа страна“ (Protegrity, nCipher, Application Security, и др.) или се може остварити коришћењем енкрипционих способности које се налазе у системима за управљање базама података (СУБП) које нуде њихови продавци (Oracle, MS SQL Server 2005, и др.). Заједничко за продавце „треће стране“ и „*native*“ продавце СУБП је следеће [7]:

- Оба решења фокусирају се на енкрипцију и декрипцију на нивоу базе података.
- Оба решења се ослањају на аутентификацију и контролу приступа СУБП да би чували и повлачили енкриптоване податке. Мада неки продавци „треће стране“ нуде додатне способности аутентификације и контроле приступа.
- Оба решења нуде чување кључева у бази података или у неком фајлу, док неки продавци „треће стране“ нуде способност чувања кључева на неком хардверском сигурносном модулу нпр. HSM (енг. Hardware Security Modul).
- Ни једно решење не нуди енкрипцију на нивоу слога већ само на нивоу колоне или целе табеле.

Без обзира које ће се решење користити за енкрипцију над колоном, односно табелом базе података, потребно је водити рачуна о перформансама. Постоје неке основне препоруке којих се треба придржавати, и то:

- Потребно је идентификовати осетљиве податке и енкриптовати само колоне са тим подацима.
- Никада не треба енкриптовати поља која се користе као спољни или примарни кључ или која се користе као индех.

Пре примене енкрипције потребно је истражити листу најчешћих наредби које се извршавају над базом података, пошто многе апликације користе велики број упита. Анализа употребе и фреквенције SQL наредби омогућава доношење правилне одлуку о томе како ће енкриптована колона утицати на перформансе. Утицај енкрипције на перформансе биће приказан на следећем примеру.

Пример: Табела садржи информације о купцима у следећим колонама: ИДКупца, ИмеКупца, АдресаКупца и БројКредитнеКартице. Она садржи податке од милион купаца, тј. има милон слогова. Над табелом се може извршити више начина енкрипције са предностима и последицама по перформансе за сваку имплементацију, и то:

1. Енкрипција свих колона у табели. Ако је потребно извршити SELECT над колоном ИДКупца за одређену вриедност ИДКупца, онда ће се морати извршити дешифровање колоне ИДКупца за свих милион слогова. Ово ће имати за последицу велико оптерећење по перформансе. Када се врши INSERT у табелу, оптерећење је минимално, међутим ако се врши UPDATE на колони ИДКупца, за одређену вредност ИДКупца, биће потребно поново извршити дешифровање за сваки појединачни слог.

2. Енкрипција три колоне у табели. У овом случају није енкриптована колона ИДКупца. Када се врши SELECT над колоном ИДКупца за одређену вриедност ИДКупца, не захтева се обављање дешифровања све док се тражени слог не поклопи са одабраним критеријем. Време које је потребно за обављање операције SELECT је минимално, али ће дешифровање пронађеног слога допринеети релативно малом оптерећењу перформанси. Уколико се уместо претраживања по колони ИДКупца одабере претраживање по колони ИмеКупца, онда ће се морати обавити дешифровање свих милион слогова за колону ИмеКупца док се не пронађе одговарајући слог, што ће за последицу имати спор упит. Исто ће се десити и код операција UPDATE и DELATE. Уколико се одабере SELECT над колонама ИДКупца и ИмеКупца, за одговарајућу вредност ИДКупца, претраживање ће бити незнатно брже зато што ће се дешифровање извршити у колони ИмеКупца само за слог који одговара траженој вриедности ИДКупца. Наравно, уколико се одабере прво претраживање на основу вредности ИмеКупца пре ИДКупца, резултат ће бити потпуно другачији, пошто ће се прво морати извршити дешифровање података у колони ИмеКупца, а онда пронаћи тражена вредност.

3. Енкрипција само колоне БројКредитнеКартице. Шансе да упит може значајно да успори рад базе података смањују се уколико се енкриптује само број кредитне картице. Значајни удар на перформансе ће бити само ако се одлучи да се претражује по колони БројКредитнеКартице. Ово ће се десити само ако се тражи одређени број кредитне картице без обезбеђивања било којег другог критерија претраге.

Осим осигурања података на нивоу колоне, овај начин енкрипције осигурава податке и на нивоу фајла, којег базе података користе за чување информација. Енкрипција на нивоу базе података транспарентна је у односу на апликациони ниво, а то значи да елиминише захтеве за измену на ниво модела апликације и указује на повећан тренд уградње пословне логике унутар система за управљање базама података коришћењем ускладиштених процедура и окидача. Пошто се енкрипција/декрипција дешава унутар базе података, ово решење не захтева познавање карактеристика апликација које приступају енкриптованим подацима. Међутим, захтева рад на интеграцији на нивоу базе података, укључујући измене на постојећим шемама база података, односно на коришћеним окидачима и ускладиштеним процедурама за узимање функција енкрипције и декрипције. Додатно је потребно пажљиво размотрити и утицај на перформансе и на хардверске способности. Недостатак овог приступа је да су подаци у форми отвореног текста између апликације и базе података и узрокује оптерећење перформанси. Предности овог приступа су: транспарентност у односу на апликацију, мало време примене, подаци су енкриптовани на диску или медију за *backup* података, а трошкови одржавања су минимални.

2.3.6.7. Шифровање фајла

Оснивни фајлови, у које база података складишти податке, су основна тачка напада. Овај ризик је повећан, због тога што се фајлови база података непрестано реплицирају на положаје за опоравке од катастрофа и праве се резервне копије (енг. backup) на преносиве медије који се касније могу оставити без надзора. Било који администратор система или спољни нападач који оствари приступ систему, може украсти податке из ових фајлова. Један основни напад на фајлове база података је отворити их са неким едитором и потражити потребне информације. Сложенији напад је коришћење алата да се претраже бинарни подаци и истражи фајл да се одреди његова структура и формат. Софистицирани нападачи могу поставити фајлове података на своје инстанце база података и тако остварити пун приступ свим подацима у њима. На срећу, енкрипција решава ове проблеме. Енкрипција на нивоу фајла, односно на нивоу *storage*, штити осетљиве податке у фајловима база података од свих нападача који немају кључ за енкриптовање [7].

Предности ове врсте енкрипције су: штити податке од напада са нивоа оперативног система и смањује ризик од физичких напада, релативно је једноставна имплементација, кратко време примене, лако се одржава. Недостатци су: не може спречити нападе на нивоу базе података какви су SQL *injection* и напади DBA, подаци су у форми отвореног текста на нивоу између система за управљање базама података и апликације, не могу се криптовати појединачни диелови фајла већ само цели фајл. Енкрипција на нивоу фајла је решење енкрипције типа све или ништа, односно нема селективног приступа – једном када се приступ оствари сви подаци су изложени осим ако нису претходно енкриптовани на нивоу апликације или базе података.

2.3.6.8. Транспарентно шифровање

Транспарентна енкрипција је енкрипција која се аутоматски примењује при свакој промени или уносу потенцијално осетљивих података. Аутоматска примена енкрипције значи да нема потребе за експлицитним позивањима енкрипцијских функција. Тиме се избегавају програмске грешке као што је позивање погрешних енкрипцијских функција, повезивање исправних функција, али са погрешним параметрима или њихово неповезивање у случајевима када је то потребно. Ово се постиже премештањем енкрипције са нивоа апликацијског слоја на слој базе података. Термин транспарентна енкрипција се односи и на то, да је она транспарентна само ауторизованим корисницима, али не нападачима. Сигурност система са транспарентном енкрипцијом зависи од тога како се успешно разликују легитимни корисници од нападача Oracle (Oracle Database 10*g* Release 2) поседује способност транспарентне енкрипције [7].

2.3.6.9. Управљање кључевима

Примена криптографије уноси и одређене ризике. Један од очитих ризика је губитак кључа. Било да је кључ изгубљен, корумпиран или избрисан резултат може бити губитак података који су енкриптовани тим кључем. Ова приетња је један од разлога зашто је управљање кључевима битна тема у криптографији. Слабо управљање кључевима може угрозити целокупну сигурност. Ако нападач може приступити кључу он га може искористити за декриптовање података. Према томе, управљање кључевима треба да да одговоре на више питања, као што су: како се приступа кључевима, како се њима управља, где су лоцирани, како се може направити њихова резервна копија и како се они могу опоравити (енг. recovery), како се може обезбедити њихов интегритет, како се врши надзор и евидентира приступ. Кључеви се могу чувати од свих претњи коришћењем хардверског сигурносног модула HSM (енг. *Hardware Security Modul*), а ако HSM не постоји онда кључеве треба чувати одвојено од базе података [7].

2.3.7. Регистар и ревизија корисникових акција (Audit Trail)

Неопходно је претпоставити да систем сигурности базе података није савршен. То значи да је могуће да он понекад буде „разбијен“. Због тога је неопходно водити посебан LOG који се назива „Регистар корисникових акција“ и повремено вршити преглед (ревизију) овог регистра. Запис у регистру садржи [7]:

1. изворни текст корисниковог захтева,
2. идентификатор терминала са којег је захтев постављен,
3. идентификатор корисника који је поставио захтев,
4. датум и време,
5. релације, н-торке и атрибути којима је приступљено,
6. нову и стару вредност.

3. Сигурносне копије и опоравак база

3.1.Сигурносне копије

3.1.1. Потреба за израдом сигурносних копија

За време рада СУБП-а разне грешке могу узроковати недоступност база или оштећење података у њима. Честа појава је да због хардверске грешке откаже диск па да због тога подаци постану недоступни. Могућ је и квар на меморијским чиповима, матичној плочи или некој другој хардверској компоненти, што за последицу може имати и пад целог система Софтверске грешке такође могу узроковати проблеме у функционисању СУБП-а.То могу бити интерне грешке у самом СУБП-у, грешке у оперативном систему или у неком другом софтверу који је повезан с СУБП-ом. Иако оне обично не узрокују оштећења података, може се догодити да због њих цела СУБП инстанца постане недоступна или да поједини делови система не раде исправно.

Најчешће су људске грешке, као на пример погрешно ажурирање или нехотично брисање података од стране крајњих корисника, али и од стране администратора базе. Администратор може покренути неке скрипте у погрешно време или са кривим параметрима па на тај начин покварити податке. Што се пре настале грешке уоче и исправе, штета ће бити мања.

Задатак администратора база је враћање базе у стање пре настанка проблема и омогућавање нормалног наставка рада. Тај се поступак назива опоравком (енг.***RECOVERY***). Да би се опоравак база могао извести, треба имати копије база из времена пре него се грешка догодила. Такве копије, из којих се базе у случају потребе могу опоравити, зову се ***резервне копије база*** *(енг. backups) [6]*.

3.1.2. Типови сигурносних копија

За разлику од копирања неких "обичних "датотека, попут слика, мултимедијалних или текстуалних датотека, израда сигурносних копија база је сложенији задатак. Базе могу бити велике и њихово потпуно копирање може трајати јако дуго. Због тога ће се понекад израђивати копије само оних података који су се променили. будући да се базе састоје од data и log датотека, треба знати како ускладити копирање тих двеју врста датотека да би се базе из својих копија могле вратити у конзистентно стање. Да би се омогућило време опоравка унутар граница које захтева пословање, а да се због израде копија не ремети нормалан рад система, треба знати који типови копија стоје на располагању, које су њихове карактеристике и како се користе. Тада се може направити добра стратегија израде сигурносних копија базе које се администрира.
Сваки СУБП има своје специфичности па се тако и процес израде резервних копија и врсте копија разликују од једног до другог СУБП-а.

- **On-line и off-line сигурносне копије**

Сигурносне копије се могу израђивати док је база он-лине, односно за време док су корисници повезани на базу и изводе по њој читања или записивања. Код таквог копирања потребно је имати забележене и промене које су се над подацима догађале док је копирање трајало. Те су промене записане у лог датотеци. Приликом опоравка базе из on-line резервне копије, СУБП ће након враћања података из датотека морати прочитати копиране лог податке и поновити трансакције које су потврђене (енг. committed), односно поништити оне непотврђене (енг.uncommitted). На тај се начин осигурава конзистентност података након опоравка. Често се овај тип резервне копије назива hot backup.

Сигурносна копија може се направити и након што се база стави off-line. Приликом стављања off-line, СУБП се побрине да подаци у тако "спуштеној" бази буду конзистентни. Након тога, довољно је копирати само датотеке са подацима јер за време копирања нема никаквих промена над базом.Опоравак из off-line копија је бржи, али понекад пословање захтева сталну доступност па таква копирања нису изводљива. Овакве копије понекад се називају cold backups.

- **Потпуне и инкременталне резервне копије**

Потпуна сигурносна копија (енг. full backup) садржи све странице свих датотека базе. За разлику од ње, инкрементална копија (енг. incremental backup) садржи само странице пренос датотека промењене након креирања последње потпуне или инкременталне копије. СУБП-и се разликују по томе које могућности код креирања инкременталних копија нуде. Једнако тако се разликују и називи који се употребљавају. У Oracle-у, на пример, постоје две врсте инкременталних копија - диференцијалне и кумулативне.

Диференцијалне садрже само промене настале након задње инкременталне копије, а кумулативне садрже промене настале након задње потпуне копије. За разлику од тога, СКЛ Сервер има само једну врсту инкременталних копија које Microsoft назива диферецијалнима, а оне садрже све промене настале од задње потпуне копије. Кад су базе велике, обично се појављује проблем с трајањем потпуног копирања. Да би се он заобишао, израђују се инкременталне копије, што може трајати знатно краће. Но, ако треба вратити базу која је била инкрементално копирана, такав ће опоравак дуже трајати јер је потребно прво рестаурирати базу из потпуне копије, а онда на њу применити промене из инкременталне копије.

У примеру са *Слике10.* потпуна копија базе ради се понедељком у 3.00. Након тога, уторком, средом и четвртком израђују се инкременталне копије у исто време. Ако у четвртак у 10.00 база падне, опоравак ће се израдити прво рестаурирањем базе из потпуне копије од понедељка, а након тога ће се применити промене из инкременталне копије од четвртка (зависно о типу инкременталних копија, може бити потребно рестаурирати промене из свих инкременталних копија редом).

Слика10.*Распоред израде потпуних и инкременталних копија [6]*

Инкрементално копирање погодно је кад су базе велике, али у њима нема пуно промена података. Тада уштеда на времену копирања може бити пуно важнија него незнатно повећање времена за опоравак. Код малих база, код којих потпуно копирање не траје дуго, није потребно радити инкременталне копије јер се на тај начин усложњава процедура копирања, а погодности нису велике. Неки СУБП-и имају могућност анализирања количине промена у бази па могу сами одлучити хоће ли се направити потпуна или инкрементална копија. Администратори могу подесити граничну вредност, тј проценат промењених података изнад којега ће СУБП радити потпуну копију. Уопштено, кад се промени 40% или више података, инкременталне копије нису добар избор.

Код неких СУБП-а постоје помоћни програми за спајање инкременталних копија. Више инкременталних копија може бити спојено у једну или се оне могу спојити са потпуном копијом стварајући на тај начин нову потпуну копију. Такво спајање је добро покренути одмах након израде инкременталне копије. При спајању тих копија нема проблема с ометањем нормалног рада базе, а такође смањује се време потребно за опоравак.

3.1.3. Сигурносне копије ЛОГ датотека

Свака промена података или структуре базе записује се у њену лог датотеку. СУБП користи записе из лог датотеке да би поновним извођењем или поништавањем тих промена у процесу опоравка могао довести базу у конзистентно стање. Копија лог фајлове је нужна кад је потребно базу вратити у стање какво је било у тачно одређеном тренутку у прошлости. Опоравак тада започиње из податковне копије, а наставља се читањем копије лога и понављањем промена насталих након израде податковне копије, те поништавањем оних које су потврђене након тренутка у који се треба вратити. Лог датотека у коју се подаци о променама тренутно записују назива се **активним логом**. С временом у лог датотеци има све више записа и она се може попунити до краја. Тада СУБП укључује процес архивирања лога (енг. Log Offloading) , којим се активни лог копира на другу локацију. Кад се активни лог архивира, он се ресетује и подаци се опет записују у њега. Овај концепт архивирања лога је у сваком СУБП-у нешто другачији. Тако се, на пример, код SQL Server-а не говори о архивирању нити Log Offloading-у, већ о backup-у лог фајлове. Она има активни и неактивни део. Извођењем backupa лога неактивни део се копира у backup датотеку и након тога се простор који је он заузимао ослободи за нова записивања. Обично сваки СУБП има неколико начина рада са лог фајловима, а разликују се у детаљности записивања промена те могућношћу копирања лога. SQL Server има три начина рада: simple, bulc-logged и full. Simple начин рада не даје могућност копирања лога пасе у њему не може рачунати на опоравак у било коју тачку у прошлости.

3.1.4. Копије делова базе

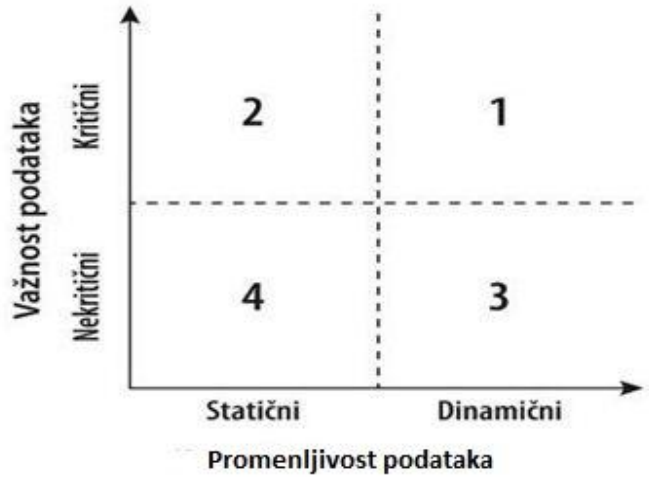
Код великих база, где су израде резервних копија дуготрајне, могу бити корисне могућности копирања само појединих делова базе. Обично се то односи на скуп физички повезаних табела. У SQLServer-у се скуп таблица може сместити у различите фајлове, а свака датотека се може засебно копирати. Ако у базама постоји пуно архивских података, који се само читају и више се не ажурирају, онда је таблице добро раздвојити тако да се оне архивске (енг. Read-only) не налазе у истим датотекама као и оне над којима се догађају промене. Тада можемо успоставити стратегију у којој 01:00 Тада се може направити потпуна копија базе, а након тога урадити копирање само оних датотека с таблицама чији подаци се мењају. Read-only податке не треба копирати тако често.

3.1.5. Одређивање распореда израде резервних копија

Код креирања распореда по којем ће се безбедносне копије за неку базу креирати, важно је направити добар баланс између две супротна захтева: оног да се копије израђују што чешће како би се смањило касније време опоравка и оног да израде копија што мање ометају нормалан рад система. Нису сви подаци у бази једнако важни. Неки могу бити пресудни за пословање компаније, док су други споредни или су доступни и на неким другим местима. Пре него се успостави стратегија и распоред за израду сигурносних копија свакако би било добро анализирати пословање те природу података и њихову важност. У тој анализи треба дати одговоре на следећа питања:

- Колико се често подаци мењају?
- Колико су подаци важни за пословање?
- Могу ли се одређени подаци лако рекреирати из неких других извора?
- Колико је лоше ако се одређени подаци изгубе?
- Постоји ли време у којем нико не треба радити са базом? Је ли потребна доступност 24сата дневно, 7 дана у недељи?
- Колико је скупо време недоступности система због опоравка?

Након такве анализе, подаци се могу класификовати према важности и промењивости, као на графу са *Слике 11*. Што се податак налази више десно, он је динамичнији, тј чешће се мења. Што се налази више горе, то је он важнији за пословање. Кад се поадци сместе на такав граф, из њега се може закључити колико их често треба копирати. Уопштено, што се подаци чешће мењају, чешће их требакопирати. Исто важи и за веома важне податке.



Слика11.*Класификација података по важности и промењљивости [6]*

Подаци из првог квадранта на графу са *Слике11.* врло су промењљиви и изузетно важни за пословање. Собзиром на то, у процесу опоравка треба их брзо вратити па их због тога треба копирати често. У другом квадранту су подаци који су од велике важности, али се не мењају често. Њих треба брзо вратити у случају пада базе. Прихватљиво решење може бити инкрементално копирање таквих података, уз спајање са потпуном копијом да би се време враћања таквих података што више смањило. Подаци из трећег квадранта често се мењају, али нису превише важни. Можда се могу рекреирати помоћу неких скрипти па, ако их има јако много, не морају се копирати. У четврти квадрант смештени су подаци који су статични и од мале важности. Њих треба укључивати у распоред копирања тек након што се тај распоред установи за податке из прва три квадранта.

3.1.6. Логичке резервне копије

За разлику од метода које су до сада споменте и у којима се датотеке базе физички копирају (па се зато често зову *image copies*), у специјалним случајевима биће згодно копирати само податке, а не и целу физичку датотеку са својом структуром. Обично се то своди на извоз података из табела података наредбом EXPORT или UNLOAD.

Корисне су у следећим случајевима [6]:

- Опоравак слога или објекта - у случају погрешног брисања;
- Прелаза на ново издање СУБП-а - Произвођачи често промене интерну структуру БП у новом издању;
- Миграција података у хетерогеном окружењу - Пренос у други СУБП, Excel табелу или слично.

3.2. Опоравак базе података

Опоравак базе података (енг.*recovery*) представља процес враћања базе података у коректно стање. Сасвим је реално, и дешава се, да услед отказа система мора да се уради опоравак базе података.

Опоравак подразумева активност коју систем предузима у случају да се, у току извршења једне или више трансакција, открије неки разлог који онемогућава њихово успешно комплетирање. Тај разлог може бити [3]:

- *у трансакцији, као што је прекорачење неке од дозвољених вредности (пад трансакције);*
- *у систему, нпр. престанак електричног напајања (пад система), или*
- *у диску на коме је база података, нпр. оштећење глава диска (пад медија).*

Како је трансакција логичка јединица посла, њене радње морају или све успешно да се изврше или ниједна, па је у случају немогућности успешног комплетирања неопходно поништити ефекте парцијално извршених трансакција. Са друге стране, може се догодити да у тренутку пада система неки ефекти успешно комплетираних трансакција још нису уписани у базу (нпр. ажурирани слогови су још у унутрашњој меморији– у бафер података чији се садржај при паду система губи); зато може постојати потреба за поновним (делимичним) извршењем неких претходно успешно комплетираних трансакција. То поништавање односно поновно извршавање у циљу успостављања конзистентног стања базе представља системску активност опоравка од пада трансакције или система. **Трансакција је, дакле, и логичка јединица опоравка.**
У случају пада медија, опоравак укључује пре свега преписивање архивиране копије базе (нпр. са траке) на исправан медиј, а затим, евентуално, поновно извршење трансакција комплетираних после последњег архивирања а пре пада медија. Овако конципиран опоравак нужно се заснива на постојању поновљених (дуплираних) података и информација на различитим местима или медијима, тј. на могућности реконструкције информације на основу друге информације, поновљено смештене на другом месту у систему. Као "друго" место у систему, на које се (осим у базу) уписују информације о извршеним радњама, обично се користи тзв. лог датотека (системски лог, дневник ажурирања)- о чему је било речи у претходном одељку.

Компонента СУБП одговорна за опоравак базе података од пада трансакције, система или медија је **управљач опоравка**. Његова активност се глобално састоји од следећих радњи:

- периодично преписује (енгл. *dump*) целу базу података на медиј за архивирање;
- при свакој промени базе података, уписује слог промене у лог датотеку, са типом промене и са новом и старом вредношћу при ажурирању, тј. са новом вредношћу при уношењу у базу и старом вредношћу при брисању из базе;
- **у случају пада трансакције или система**, стање базе података може бити неконзистентно; Управљач опоравка користи информације из лог датотеке да поништи дејства парцијално извршених трансакција односно да поново изврши неке комплетиранетрансакције; архивирана копија базе података се не користи;
- **у случају пада медија** (нпр. диска на коме је база података), "најсвежија" архивирана копија базе података се преписује на исправни медиј (диск), а затим се користе информације из лог датотеке за поновно извршење трансакција комплетираних после последњег архивирања а пре пада медија.

Да би се омогућило управљачу опоравка да поништи односно да поново изврши трансакције, потребно је обезбедити процедуре којима се поништавају односно поново извршавају појединачне радње тих трансакција којима се мења база података. База података мења се операцијама ажурирања (у ужем смислу), уношења или брисања података, па поред процедура за извршење тих операција, СУБП мора бити снабдевен и одговарајућим процедурама за поништавање односно поновно извршење тих операција, на основу старих односно нових вредности запамћених у системском логу.
Другим речима, СУБП поседује, осим тзв. **DO**-логике ("уради"), и тзв**. UNDO** ("поништи") и **REDO** ("поново уради") логику.

Пад система или медија може се догодити и у фази опоравка од претходног пада. Зато може доћи до поновног поништавања већ поништених радњи, односно до поновног извршавања већ извршених радњи. Ова могућност захтева да UNDO и REDO логика имају својство идемпотентности тј. да је UNDO(UNDO(x)) ´ UNDO(x) и REDO(REDO(x)) ´ REDO(x) за сваку радњу x.

У лог датотеци показивачима су повезани слогови који се односе на једну трансакцију. То подразумева чување лог датотеке на медију са директним приступом, нпр. на диску. С друге стране, количина података који се уписују у лог датотеку може бити веома велика (неколико стотина милиона бајтова дневно). Зато се лог датотека обично организује тако да има свој активни део на медију са директним приступом, и архивирани део (нпр. на траци), у који се преписује активни лог кад год се препуни. Претпоставља се да лог датотека никада, или веома ретко, "пада", тј. да је, захваљујући сопственом дуплирању, триплирању, итд, на разним медијима, увек доступна.

3.2.1. Опоравак од пада трансакције

Један апликативни програм може се састојати од већег броја трансакција. Извршење једне трансакције може да се заврши планирано или непланирано. До планираног завршетка долази извршењем **COMMIT** операције којом се успешно комплетира трансакција, или експлицитне **ROLLBACK** операције, која се извршава када дође до грешке за коју постоји програмска провера (тада се радње трансакције поништавају а програм наставља са радом извршењем следеће трансакције). Непланирани завршетак извршења трансакције догађа се када дође до грешке за коју не постоји програмска провера; тада се извршава имплицитна (системска) ROLLBACK операција, радње трансакције се поништавају а програм прекида са радом.

Извршавањем операције COMMIT, ефекти свих ажурирања трансакције постају трајни, тј. више се не могу поништити процедуром опоравка. У лог датотеку се уписује одговарајући слог о комплетирању трансакције (COMMIT слог), а сви катанци које је трансакција држала над објектима – ослобађају се. Извршење COMMIT операције не подразумева физички упис свих ажурираних података у базу (неки од њих могу бити у бафер података, при чему се, ефикасности ради, са физичким уписом у базу чека док се бафери не напуне). Чињеница да ефекти ажурирања постају трајни значи да се може гарантовати да ће се упис у базу догодити у неком наредном тренутку; у случају пада система, на пример, после COMMIT операције а пре физичког уписа података из бафера у базу, упис се гарантује REDO-логиком.

Пад трансакције настаје када трансакција не заврши своје извршење планирано. Тада систем извршава имплицитну – принудну ROLLBACK операцију, тј. Спроводи активност опоравка од пада трансакције. Појединачне радње ажурирања у оквиру једне трансакције, као и операција почетка трансакције (експлицитна или имплицитна BEGIN TRANSACTION операција), извршавају се на начин који омогућује опоравак базе у случају пада трансакције. Опоравак података у бази и враћање базе у конзистентно стање омогућује се уписом свих информација о овим радњама и операцијама у системски лог. При извршавању операција ажурирања (у ужем смислу), осим што се ажурира база, у лог се бележе вредности сваког ажурираног објекта пре и после ажурирања (уз остале информације о том ажурирању).
Извршавањем операције BEGIN TRANSACTION (било да је експлицитна или имплицитна) у лог датотеку се уписује слог почетка трансакције. Операција ROLLBACK, било експлицитна или имплицитна, састоји се од поништавања учињених промена над базом. Извршава се читањем уназад свих слогова из лог датотеке који припадају тој трансакцији, до BEGIN TRANSACTION слога. За сваки слог, промена се поништава применом одговарајуће UNDO-операције. Активност опоравка од пада трансакције не укључује REDO логику.

3.2.2. Опоравак од пада система

У случају пада система, садржај унутрашње меморије је изгубљен. Зато се, по поновном стартовању система, за опоравак користе подаци из лог датотеке да би се поништили ефекти трансакција које су биле у току у тренутку пада система. Ове трансакције се могу идентификовати читањем системског лога уназад, као трансакције за које постоји BEGIN TRANSACTION слог али не постоји COMMIT слог. Да би се смањила количина посла везана за поништавање трансакција, уводе се, у правилним интервалима, обично после одређеног броја уписаних слогова у лог датотеку, **тачке памћења**. У моменту који одређује тачку памћења, физички се уписују подаци и информације из лог бафера и бафера података (који су у унутрашњој меморији) у лог датотеку и у базу података, REDOм, и уписује се слог тачке памћења у лог датотеку. Овај слог садржи информацију о свим активним трансакцијама у моменту тачке памћења, адресе последњих слогова тих трансакција у лог датотеци, а може садржати и низ других информација о стању базе података у том тренутку. Адреса слога тачке памћења уписује се у датотеку поновног стартовања (енгл. *RESTART FILE*).

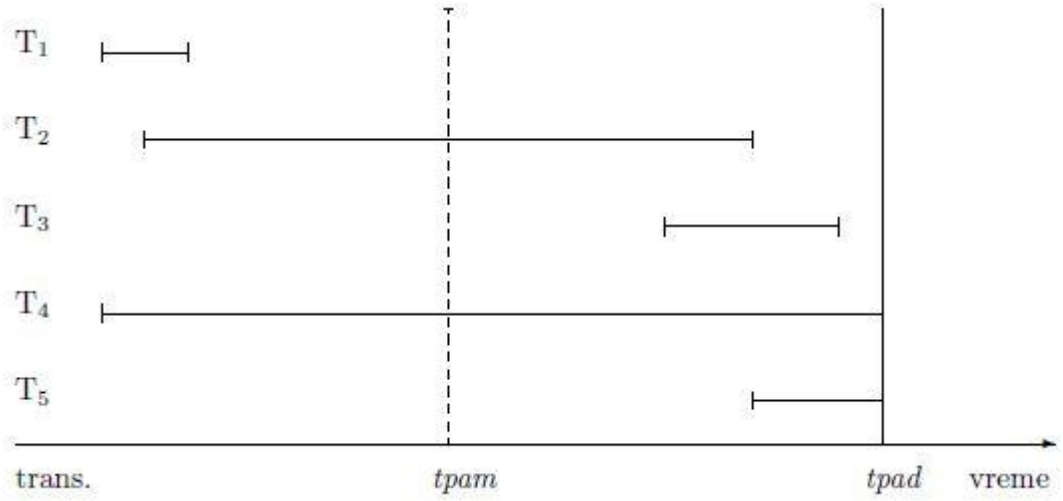
Физички упис у тачки памћења значи да ће се садржај бафера преписати на спољни медиј без обзира да ли су бафери пуни или не. То даље значи да се физички упис свих ажурираних података успешно комплетиране трансакције може гарантовати тек у тачки памћења која следи за успешним комплетирањем трансакције. Тако се долази до закључка да су комплетирање трансакције (упис COMMIT слога у лог датотеку) и дефинитивни упис свих ажурирања те трансакције у базу – две одвојене радње, за које се не сме догодити да се једна изврши а друга не. Механизам који то обезбеђује је протокол **унапредног уписивања у лог** (енгл. *WAL – WRITE AXEAD LOG*).

Према овом протоколу, при извршењу операције COMMIT прво се одговарајући слог физички уписује у лог датотеку, па се затим подаци уписују из бафера података у базу. Ако дође до пада система после уписа COMMIT слога у лог датотеку а пре него што је садржај бафера података преписан у базу, тај се садржај може рестаурисати из системског лога REDO логиком.

При поновном стартовању система, после пада система, могуће је према садржају лог датотеке идентификовати неуспеле трансакције – кандидате за поништавање, и успеле трансакције – кандидате за поновно извршавање. Опоравак базе података тада се врши према протоколу који се може описати следећим псеудопрограмом :

BEGIN
наћи слог последње тачке памћења у лог датотеци
(из датотеке поновног стартовања);
IF у последњој тачки памћења нема активних трансакција
и слог тачке памћења је последњи слог у лог датотеци,
опоравак је завршен
ELSE

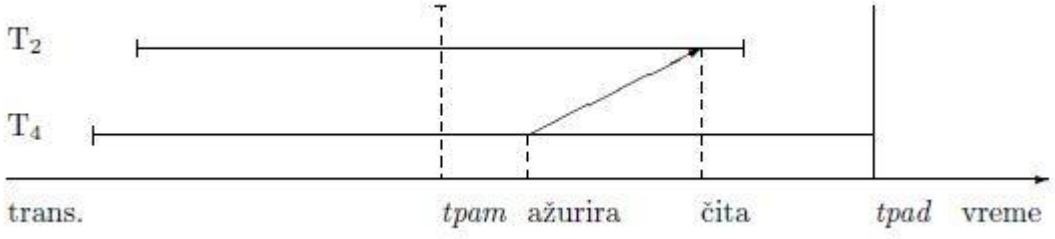
BEGIN
формирати две празне листе трансакција, “успеле” и “неуспеле”;
све трансакције активне у последњој тачки памћења
ставити у листу неуспелих;
читати REDOм лог датотеку од тачке памћења до краја:
када се наиђе на слог “почетак трансакције”,
додати трансакцију листи неуспелих;
када се наиђе на слог “комплетирана трансакција”,
додати (преместити) ту трансакцију листи успелих;
читати уназад лог датотеку (од краја)
и поништити акције и ефекте неуспелих трансакција;
читати лог датотеку од последње тачке памћења унапред
и поново извршити успеле трансакције
END
END.



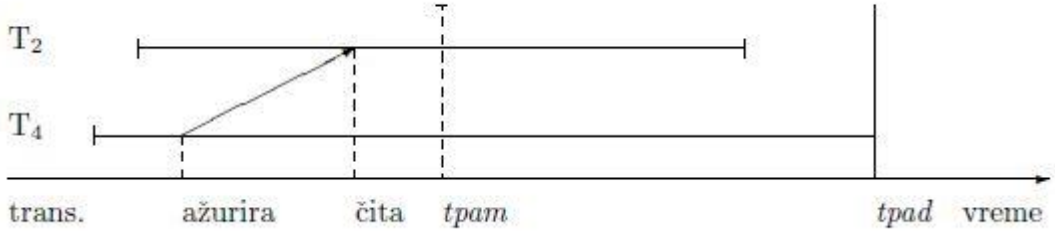
Слика12. *Пад система при извршењу скупа трансакција [3]*

На временској оси (*Слика12.*) означене су две тачке: tpam – тачка памћења, и tpad – тачка пада система. Трансакција T1 завршила је рад пре последње тачке памћења, па су њена ажурирања трајно уписана у базу активношћу постављања те тачке, и зато она не улази у процес опоравка. Трансакције T2 и T3 завршиле су са радом после момента tpam а пре пада система. У моменту пада система активне су трансакције T4 и T5. Због тога претходни протокол сврстава трансакције T4 и T5 у листу неуспелих (чија дејства поништава), а трансакције T2 и T3 у листу успелих (које затим поново извршава од тачке tpam). Тиме је завршена активност опоравка од пада система.

Управљач опоравка подразумева да трансакција ослобађа све X-катанце приизвршењу COMMIT операције. Најчешће је исти случај и са S-катанцима. Ово је сагласно са својством изолације трансакције и решава проблем зависности од поништеног ажурирања, било да се оно догодило пре или после тачке памћења. Дакле, нису могућа извршења приказана на *Слици 13. и Слици 14.*



Слика13. *Зависност од поништеног ажурирања после тачке памћења [2]*



Слика14. *Зависност од поништеног ажурирања пре тачке памћења [2]*

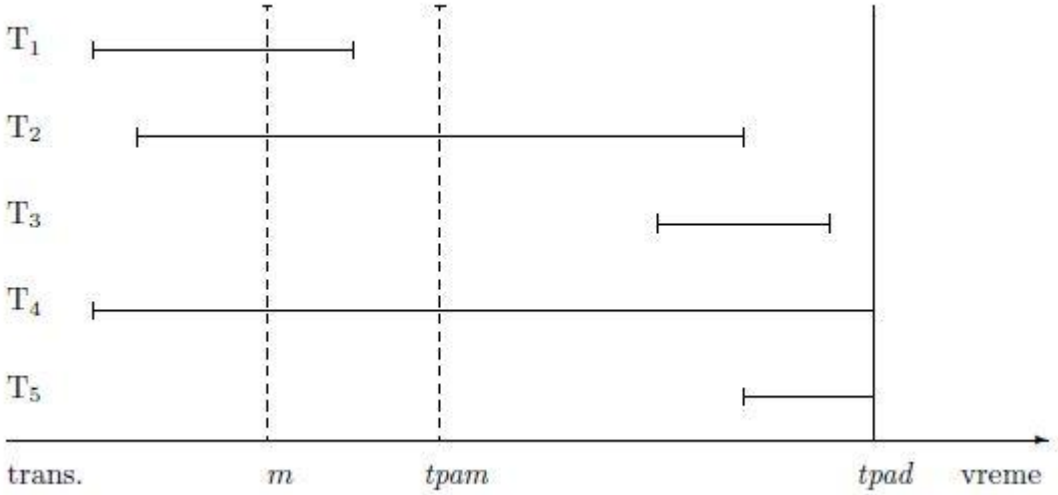
3.2.3. Побољшање процедуре опоравка од пада система

Постоји низ побољшања и модификација изложеног концептуалног поступка опоравка од пада система. Тако је могуће сва уписивања једне трансакције у базу оставити за тренутак извршења COMMIT операције те трансакције, чиме се елиминише потреба за UNDO логиком. Такође је могуће ослабити захтев за изолацијом трансакције, тј. захтев за двофазношћу, што носи опасност нелинеаризованог извршења. Једно евидентно побољшање протокола опоравка од пада система односи се на активности везане за тачку памћења. Наиме, претходно описани протокол поништава ефекте неуспелих трансакција који можда нису ни били убележени у базу (већ само у бафер података ако су се догодили после тачке памћења), односно поново извршава радње успелих трансакција од тачке памћења, чак и ако су ефекти тих радњи, због попуњености бафера података, уписани у базу пре пада система [2].

Могуће је елиминисати физичко уписивање бафера података у базу података у тачки памћења, а у фази опоравка од пада система поништавати само оне радње неуспелих трансакција чији су ефекти уписани у базу, односно поново извршавати само оне радње успелих трансакција чији ефекти нису уписани у базу.

У том циљу уводи се, у сваки слог системског лога, у време његовог уписа у лог, јединствена ознака – ***серијски број у логу*** (енгл. *LSN – LOG SEQUENCE NUMBER*). Серијски бројеви су у растућем поретку. Осим тога, кад год се ажурирана страница физички уписује у базу података, у страницу се уписује и LSN слога системског лога који одговара том ажурирању. У системском логу може бити више слогова који одговарају ажурирању исте странице базе података. Ако је серијски број уписан у страницу П већи или једнак серијском броју слога лога, ефекат ажурирања коме одговара тај слог физички је уписан у базу; ако је мањи, ефекат није уписан. Да би се слог из базе података ажурирао, потребно је прочитати (из базе) страницу на којој се слог налази. После ажурирања слога, страница је (у баферу података) спремна за упис у базу, при чему и даље носи серијски број свог претходног уписа у базу. Сада се у процедуру регистровања тачке памћења, осим што се елиминише физички упис бафера података у базу, може додати упис, у слог тачке памћења, вредности LSN m најстарије странице бафера података, тј. најмањег LSN страница из бафера података, које су ажуриране али још нису уписане у базу. Слог системског лога са серијским бројем m одговара тачки у системском логу од које треба, уместо од тачке памћења, при опоравку од пада система поново извршавати успеле трансакције. Како тачка m претходи тачки памћења, може се десити да нека трансакција која је успешно комплетирана пре тачке памћења, "упадне" у скуп активних (успелих) трансакција (трансакција T1, *Слика15*). На такву трансакцију примениће се процедура опоравка, мада се то не би догодило у претходној варијанти опоравка**;** то је "цена" смањеног броја поништавања, поновног извршавања и физичког уписа које обезбеђује овај поступак.

Процедура опоравка од пада система сада има следећи изглед:



Слика15. *Опоравак од пада система коришћењем серијских бројева [2]*

BEGIN
формирати две празне листе трансакција, “успеле” и “неуспеле”;
све трансакције активне у тачки м ставити у листу неуспелих;
читати редом лог датотеку од тачке м до краја:
када се наиђе на слог “почетак трансакције”,
додати трансакцију листи неуспелих;
када се наиђе на слог “комплетирана трансакција”,
додати (преместити) ту трансакцију листи успелих;
читати уназад лог датотеку (од краја)
и поништити ефекте оних радњи неуспелих трансакција
за које је п ¸ р, где је р – LSN текућег слога
а п – LSN одговарајуће странице базе података;
читати лог датотеку од тачке м унапред
и поново извршити оне радње успелих трансакција
за које је п < р (п; р – као у претходном кораку)
END.

4. Закључак

Базе података са гледишта безбедности неисцрпна је и увек актуелна тема. Сваки већи посао, као на пример државне установе, ослањају се на базе података у којима спремају кључне податке. Управо због тога врло је важно базе података заштити и њима управљати на прави начин.

Напади на базе података, из године у годину, су у порасту, чиме је ризик од компромитовања све већи. Данас провајдери финансијских услуга и здравствене организације морају строго поштовати законе о приватности података. Корисници ових услуга због забринутости од откривања података или злоупотребе ће неминовно проширити одговорности сваке организације за пружање одређених услуга. Негативни ефекти, настали као последица напада, огледали би се у виду законске одговорности, негативног публицитета, изгубљеног поверења у јавности, те губљење новца и продуктивности.

Као главне препоруке за чување сигурности базе података су стална надоградња програмских пакета, одвајање базе на сигурне сегменте мреже, коришћење енкрипције при трансферу и складиштењу осетљивих података, коришћење ауторизације аутентификације и улога. Битно је не занемарити и следеће принципе:

- Шифроване податке треба држати одвојено од криптографских кључева;
- Неопходна је квалитетна аутентификација која ће бити коришћена за идентификацију корисника пре дешифровања битних података;
- Приступи кључевима треба да буду строго надгледани, као и анализирани лог фајлови;
- Битни подаци морају бити шифровани и када се налазе у комуникационом каналу између базе података иапликације.

Постоје различити начини напада на базе података, који су већ добро познати, али због недостатака у другим системима могу погодити и СУБП и базу података којом се управља. Осим недостатака других система, постоје рањивости које укључују људске факторе као што су додељивање акредитације злонамерним корисницима, немарност администратора при надгледању база података и сл.

Иако су базе података рањиве на наведене спољне и унутрашње претње, могуће је смањити број рањивости на прихватљив ниво ризика. То се постиже тако да се користе напредни сигурносни механизми описани у претходним поглављима, константно надограђују програмски пакети везани уз оперативни систем и СУБП, користе сигурни мрежни ресурси те сигурносни производи као што су firewall и антивирусни алати.

Литература

[1] Лазаревић Бранислав, Зоран Марјановић, Ненад Аничић, Слађан Бабарогић: Базе података, ФОН Београд, Београд 2003.

[2] Павловић-Лажетић Гордана: Основе релационих база података, Математички факултет, Београд, 2000.

[3] Милан Ерић: Пројектовање информационих система и база података-скрипта, Машински факултет у Крагујевцу.

[4] Младен Веиновић, Игор Франц, Александар Јевремовић: Базе података, практикум, Факултет за пословну информатику, Универзитет Сингидунум у Београду, 2006.

[5] Michael Lee, Gentry Bieker: Mastering SQL Server 2008, Wiley Publishing, Inc., Indianapolis, Indiana, 2009.

[6] Дамир Корбар: Администрирање база података - приручник, Загреб 2010.

[7] Бранислава Цвијетић, Драгана Аврам, Миљана Братић: Употреба криптографије за сигурност база података, Државна агенција за истраге и заштиту, Сарајево 2008.

[8] http://studenti.rs/skripte/baze-podataka-praktikum/

[9] http://www.crnarupa.singidunum.ac.rs/valjevo/2008 2009/Poslovna%20informatika/Informatika.pdf

[10] http://spencerwade.wordpress.com/2013/02/05/network-security-in-a-nutshell/

[11] http://www.datasolutions.rs/srp/spasavanje-podataka/spasavanje-podataka-tehnologija.html

[12] http://www.paragraf.rs/propisi/zakon_o_autorskom_i_srodnim_pravima.html

[13] http://www.rfzo.rs/download/pravilnici/drugidrzorg/CR.pdf

[14] http://www.slideshare.net/ibmsrbija/procena-it-kontrola-i-upravljanje-rizicima-informacionih-sistema