**Duomenų bazių valdymo egzamino klausimai**

**1.**

* **Duomenų bazių valdymo sistemos (DBVS) sąvoka.**

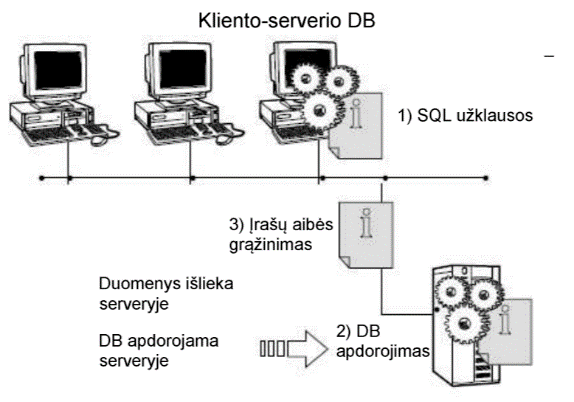
**Duomenų bazių valdymo sistema (DBVS, Database Management System) –** tai programinių priemonių kompleksas, leidžiantis:

* Aprašyti (sukurti) duomenų struktūras;
* Įvesti duomenis (aktualaus būvio palaikymas);
* Duomenų paieška (svarbiausia funkcija);
* Įvairių ataskaitų sudarymas.

**Papildomos funkcijos:**

* Duomenų vientisumo palaikymas;
* Duomenų neprieštaringumo kontrolė;
* Duomenų apsauga;
* Kelių nepriklausomai dirbančių vartotojų vienalaikis darbas su duomenimis;
* Duomenų bazės būsenos atstatymas po avarijų.
* **DBVS veiksena kliento ir serverio architektūroje, jos komponentai.**

**DB serveris yra atsakingas už:**

* Duomenų keitimo ir išgavimo apdorojimą;
* ******Duomenų apdorojimą;
* DB taisyklių ir ribojimų įgyvendinimą ir palaikymą;
* Duomenų saugumo palaikymą.

**DB klientas yra atsakingas už:**

* Duomenų pateikimą vartotojui;
* Sąsajos užtikrinimą;
* Užklausų perdavimą serveriui.

***Dažnai naudojamos trijų ir daugiau sluoksnių kliento ir serverio architektūros, kur verslo logika, įgyvendinama verslo taisyklių pagalba, yra apdorojama atskiro serverio.***

* **Pagrindiniai MS SQL Server DBVS komponentai ir jų funkcionalumas.**
* **SQL Server užklausų optimizavimo posistemis** - atsakingas už užklausų optimizavimą pagal jų operacijų kaštus.
* **Laikinosios atminties valdymo posistemis (Buffer Manager)** - atsakingas už dažnai naudojamų duomenų analizę, kurios pagalba tokie duomenys yra patalpinami į lengviau prieinamą laikinąją atmintį, taip neapkraunant serverio.
* **Lazy Writer** - įrašo pakeistus duomenis į duomenų failus.
* **Išteklių valdymo posistemis (Resource Monitor)** - optimizuoja užklausų vykdymo planus ir pašalina senus užklausų vykdymo planus.
* **Užraktų valdymo posistemis (Lock Manager)** - dinamiškai valdo užraktus.

**2.**

* **Kiekybinės ir kokybinės DBVS charakteristikos.**

**Kiekybinės:**

* Palaikomi duomenų ir jų indeksų tipai;
* Šifravimo algoritmai;
* Laikinos ir virtualios lentelės;
* Maksimalus užklausų skaičius.

**Kokybinės:**

* Apimtis;
* Taikymo sritis;
* Funkcionavimo bazė;
* Darbo patogumas.
* **Populiariausių reliacinių DBVS palyginimas.**

Visos populiariausios DB (*Oracle, MSS PostgreSQL*) turi pagrindinius funkcionalumus, išskyrus **Informix ir MySQL**, kurie negali atlikti *intersect* ir *except*.

**3.**

* **Sisteminis DB katalogas.**

**Katalogas** – saugomos duomenų bazės schemos: duomenys apie duomenis arba metaduomenys.

Kadangi katalogo lentelės skaitomos dažnai, jos turi būti optimizuotos skaitymui, dažnai geras katalogas gali būt sukurtas tik spec. priemonėmis (saugant duomenis ne įprastų lentelių pavidalu).

* **Kataloge saugoma informacija, jos kaupimas ir panaudojimas.**

**Saugomi duomenys:** lentelių, laukų vardai, domenai, PK, FK, AK, ribojimai, apsaugos duomenys ar autorizacija.

**Sisteminį katalogą naudoja įvairios DB sistemos dalys:**

* + ***DDL kompiliatoriai DB schemos sudarymo ir išsaugojimo*** ***metu*** - ribojimai ir kiti elementai yra ištraukiamis iš DDL ir SDL, tikrinami ir įrašomi į katalogą (užpildo sisteminio katalogo DB duomenimis);
  + ***Užklausų ir DML kompiliatoriai – i***šverčia SQL sakinius į žemo lygio duomenų manipuliacijos komandas. Tuo metu surandami vidinės DB schemos atitikmenys koncepciniai DB schemai.
  + ***Užklausų ir DML optimizatorius –*** nuskaito -> analizuoja geriausią būdą, optimaliausią būdą.
  + ***Užklausų ir DML sintaksės nagrinėtojas ir tikrintojas*** – tikrina ar užklausoje naudojami laukai realiai egzistuoja DB;
  + ***Autorizacijos ir saugumo patikros moduliai –*** tikrinama ar paminėtas vartotojas turi teisę manipuliuoti nustatytais duomenimis;
  + ***Išorinio*** – koncepcinio lygmenų atitikmenų nustatymas.

**Katalogo duomenys naudojami praktiškai visų sistemos modulių.**

**Informacija kataloge turi būti atnaujinama automatiškai.**

**4.**

* **Užklausos vykdymo tvarka serveryje.**

**Atpažinimo etapai:**

* **Skenavimas** (atpažįsta konstrukcinius kalbos elementus sakinyje);
* **Nagrinėjimas** (nagrinėja užklausos sintaksę ir jos korektiškumą);
* **Patikra** (tikrina ar visos lentelių ir laukų pavadinimai turi prasmę).

**POTO** sukuriama vidinė užklausos struktūra medžio/grafo pavidalu.

Nustatoma duomenų apdorojimo strategija, čia užklausos generatorius gali dirbti dvejais režimais:

* **Interpretatorius** (kodą iškart pateikia procesoriui).
* **Kompiliatorius** (išsaugo kodą ir vėliau esant reikalui apdoroja).

***Užklausos vykdymas DB procesoriumi gali būti nesėkmingas. Vietoje rezultato procesorius pateikia klaidos pranešimą.***

**SQL užklausa** - skenavimas, nagrinėjimas, patikra -> vidinė užklausos forma -> užklausos optimizavimas –> vykdymo planas -> užklausos kodo generavimas –> vykdymo kodas - >užklausos vykdymas db procesoriumi -> užklausos rezultatas.

**5.**

* **Objektinis požiūris į DB projektavimą.**

Objektiškai projektuojant DB, gaunamas rezultatas - klasių diagrama.

* **UML ir pagrindinės UML naudojamos diagramos.**

Naudojama **UML** – modeliavimo ir specifikacijų kūrimo kalba, skirta specifikuoti, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus.

UML gauta sujungus tris modelių žymėjimo būdus: **OMT, OOSE, BOOCH.**

**Diagramos** – veiklos, užduočių, bendradarbiavimo, būsenų, klasių, komponentų, panaudos atvejų.

* **Objektinio modeliavimo etapai.**

1. **Koncepcinis** (modeliuoja verslo užduotis);
2. **Loginis** (specifikuoja reikalavimus, analizę ir grubus projektavimas);
3. **Fizinis** (detalus projektavimas).

**6.**

* **Verslo sistemos analizė.**

**Daugiausiai dėmesio skiriama:** kaip darbuotojas mato verslą, kas jo nuomone padėtų kompanijai, kas daroma blogai, kaip pavieniai darbuotojai atlieka savo užduotis. (čia svarbu išsiaiškinti ir specifines verslo taisykles, išimtis).

* **Naudojami konceptai.**
* verslo aktorius;
* verslo darbuotojas
* verslo esybė
* verslo užduočių modelis
* verslo modelis.
* **Verslo užduočių modeliavimas.**

Apibendrintas išorės požiūris į verslo sistemą: verslo aktoriai.

* **Verslo objektų modelis.**

Dėmesys skiriamas verslo darbuotojams, kurie vykdo verslo procesą sistemos viduje. (kokios vyksta sąveikos su kitais verslo darbuotojais, kitomis esybėmis vykdant verslo procesą).

**7.**

* **Verslo ir sistemos modeliai.**

Verslo modelyje kalbama labiau apie esamą verslo sistemą.

Sistemos modeliavimo metu pereinama prie siekiamos būsenos ir norimos realizuoti sistemos. Čia reikalavimai – ko nori vartotojas.

* **Verslo užduočių modelio transformacija į sistemos užduočių modelį.**

IS atspindi realų pasaulį ir realius verslo poreikius, todėl galima naudoti taisykles verslo užduočių modelio transformacijai į sistemso užduočių modelį. Transformacija nėra vienareikšmė - kuriant sistemą gali būti eliminuoti kai kurie verslo sistemso darbuotojai. (verslo užduotis - posistemė, verslo aktorius – aktorius, verslo darbuotojas – aktorius/užduotis, verslo darbuotojo veikla - užduotis).

* **Užduočių aprašo šablonas.**

**Reikalavimai:** identifikuojanti informacija, apibendrintas užduoties aprašas,pagr. Užduoties vykdymo scenarijus, GALIMI elementai: ribojimai, išankstinės sąlygos, sąlygos po vykdymo, vykdymo proceso prielaidos, įterpimo ir išplėtimo taškai.

***Šablonai yra santykinai laisvi, nėra griežtos formos, paskirtis nusakoma 3-5 sakiniai.***

* **„includes“ ir ‚extends‘ stereotipai, jų naudojimas.**

**Extend** - aprašo sąveiką tarp užduočių, kai viena gali išplėsti kitą, jei tenkinamos tam tikros sąlygos.

**Include** – kai viena naudoja kitą vieną ar kelis kartus.

**8.**

* **Duomenų bazės diagrama, jos paskirtis ir pagrindiniai konceptai.**

DB diagramoje naudojamos lentelės, ryšiai, rodiniai, saugomos procedūros, domenai.

Tai yra iš pagrindo paimta klasių diagrama tačiau ji yra papildyta įvairiais plėtiniais, kurie leidžia detaliau suprasti kokio tipo yra objektai, kokias savybes turi.

* **Klasių diagramos transformacija į duomenų bazės diagramą.**

**Klasės virsta lentelėmis:** kiekvienai klasei po lentele.

**Paprasčiausias atvejis** - klasės atributas virsta lentelės stulpeliu, bendru atveju daug kas priklauso nuo to, kaip vykdoma transformacija į lenteles. (Čia jau atsiranda PK,FK,CHECK,TRIGGER,INDEX)

**9.**

**Darbo su DB duomenimis ypatumai informacinėje sistemoje.**

**Sąrašo, įrašo duomenų peržiūros formos, jų panaudojimo scenarijai ir galimybės.**

**Ataskaitų paskirtis, pagrindinės savybės ir skirtumai nuo gyvų programų sistemos formų.**

**10.**

* **Verslo analitikos (Business Intelligence) sąvoka, tradicinis požiūris.**

Terminu apibrėžiamos įvairios programų sistemos, kurios naudojamos analizuoti neapdorotiems organizacijos duomenims. BI paprastai sieja uždaviniu: duomenų gavybos,duomenų analizės,paklausimų ir ataskaitų ruošimą.

* **MS SQL Server siūlomi sprendimai.**

Turim atskiras sistemas, kurios apjungiamos pasitelkiant ETL procesą, ištraukiami reikalingi duomenys iš DB, jie patikrinami, nusprendžiama kaip juos sutvarkyti ir integruoti, tuomet sutvarkytus duomenis užkeliama į data warehouse(sandėlį), kur vyksta analizė ir gaunamos tam tikros ataskaitos.

MSSQLSERVER SPRENDIMAI: integraton,,analysis, reporting services.

**11.**

**Reporting Services sistema, jos paskirtis ir savybės.**

Tai kliento ir serverio architektūra grįsta ataskaitų generavimo sistema. Ja galima ruošti ir pateikti ataskaitas tiek spausdintame tiek virtualiuose formatuose. Reporting Services padeda kurti ataskaitų generavimo sistemas.

**Sistemos architektūra.**

**Atskaitomybės gyvavimo ciklas ir jo etapai.**

ATSKAITOMYBĖS GYVAVIMO CIKLAS: Kūrimas (sukuriamas šablonas) - Valdymas (ar duomenys gali būti gaunami ir t.t) - Pristatymas

**Duomenų apsaugos valdymas, ataskaitų prenumerata.**

DUOMENŲ APSAUGOS VALDYMAS:vartotojai turi teises į tam tikras ataskaitas (by default pagal windows sisteminę autentifikaciją), gali būti naudojama formų autentifikacija, naudojamos rolės saugumui suderinti.

**12. Ataskaitų rengimo įrankiai Visual (BI Development) Studio bei SQL Server Report Builder, jų galimybės.**

**Studio:** parenkamas duomenų šaltinis, formuojama užklausa, pasirenkamas ataskaitos tipas(lentelė, matrica), pasirenkamas elementų išdėstymas lape, pasirenkamas stilius, nurodomas ataskaitos URL.

**REPORT BUILDER:** atsiunčiamas kaip papildomas produktas, gerai tinka ruošti ataskaitas sharepoint aplinkoje, galimybė ruošti grafikus, galimas spausdinimas ar eksportas įvairiais formatais, projektas išsaugomas rdl formatu. KŪRIMAS: pasirenkamas tipas, nurodomas duomenų šaltinis, konstruojama užklausa, pasirenkamas duomenų išdėstymas, pasirenkamas stilius.

**13. ETL procesas, jo tikslai. ETL proceso etapai, fazės. Transformacijų DB.**

Extraction transofmration loading – duomenys ištraukiami iš šaltinio ir pakraunami į duomenų sandėlį.

Dažniausiai duomenys ištraukiami iš OLTP duomenų bazės, perdirbami taip, kad atitiktų duomenų sandėlio schemą ir pakraunami į duomenų sandėlio duomenų bazę.

ETL PROCESAI: extraction, transformation, loading

TRANSFORMACIJŲ DB: transformacijoms naudojama atskira DB, kuri yra kaip šaltinis duomenų sandėliai ir yra kaip paskirties vieta duomenis kurie ištraukiami iš originalios DB.

**14.**

**Duomenų saugyklų architektūros, jų elementai, pagrindinės savybės.**

Tai architektūra, kuri parodo, kad duomenys yra gaunami iš įvairių sistemų arba iš vienos pagrindinės. Struktūra, tai yra duomenys gulintis kažkokiam pavidale, kaip DB.

**Duomenų saugyklų naudojimas.**

NAUDOJIMAS: kaupiami dalykinės srities istoriniai duomenys, duomenų saugykla neprivalo apimti visų įmonės duomenų, tik strateginius, saugykla skirta specialistams, priimantiems sprendimus. Duomenų vitrina yra prieigos prie duomenų lygmuo duomenų saugyklos aplinkoje, kuri naudojama pateikiant saugyklos duomenis vartotojui, kiekviena vitrina skirta tam tikrai verslo funkcijai ar regionui.

**15.**

**Multidimensinės DB, jų savybės.**

Šiose DB duomenys yra apjungiami iš įvairių šaltinių ir vėliau naudojami sprendimų rengimui. Duomenų bazės tipas, o duomenų saugykla yra pritaikymo sritis, kurioje naudojamos multidimensinės DB. Pagrindinės multidimensinių DB taikymo sritys: duomenų saugyklos, tiesioginis analitinis duomenų apdorojimas, duomenų gavyba.

SĄVYBĖS: duomenys yra integruoti, skirtingi, laike nekinta.

**Kubas, faktas ir dimensija, jų samprata.**

KUBAS: multidimensinėse DB duomenys vaizduojami kubais, kurie leidžia apibendrinti duomenis pagal bet kokį skaičių dimensijų, kubus lengva papildyti naujom dimensijom, ląsteles sudaro dimensijų reikšmių kombinacijos. Įprastai kubas sudarytas iš vieno fakto ir daug dimensijų.

Faktas – verslo veiklos vertinimas, dažniausiai skaitinis ir adiktyvus.

Dimensija – nepriklausoma esybė dimensiniame modelyje, kuri yra kaip įeigos taškas.

**Žvaigždės ir snaigės duomenų schemos.**

ŽVAIGŽDĖ: paprasta, lengvai suprantama, geras užklausų našumas dėl nedidelio rekalingų apjungimo skaičiaus, ilgas duomenų dimensijoms užkrovimo laikas, naudojama realizuojant duomenų saugyklas.

SNAIGĖ: panaši į žvaigždės tipo duomenų schemą, bet požymiai yra normalizuoti, turi mažiau duomenų, bet yra lėtesnė.

**Užklausų tipai multidimensinėse DB.**

UŽKLAUSŲ TIPAI: pjaustymas kubeliais, agregavimo užklausos, apjungimo užklausos, eilės tipo užklausos, kubo sukimo užklausos.

**16. Išskirstytosios DB ir jų savybės.**

Kuriant išskirstytas DB duomenys saugomi ne viename kompe. Tačiau DBVS tvarkanti duomenis yra viena.

SAVYBES: duomenys fiziškai saugomi skirtinguose kompuose, nors logiškai susiję į vieną sistemą, sistemos moduliškumas užtikrina, kad sistemą galima plėsti po truputį, padidintas sistemos patikimumas parodo, kad bendra sistema veiks duotu laiko momentu, padidintas sistemos prinemumas.

**Duomenų ir verslo logikos paskirstymas.**

Modeliavimo prasme duomenis reikia teisingai išskirstyti ir numatyti duomenų mainų mechanizmus. Verslo logikos funkcionalumas gali būti tik kliento, tiek serverio dalyje ar paskirstytas, naudojami taikomųjų programų serveriai.

**Duomenų sistemų tipai.**

homogeninės(visuose mazguose veikia to paties gamintojo DBVS), heterogeninės(mazguose veikia skirtingos kilmės DBVS),

**Heterogeninės DB sistemos.**

HETEROGENINĖS DB SISTEMOS: gera integracija tarp programinių įrangų (su pilnu DBVS funkcionalumu), iš centro valdoma – visi duomenys pasiekiami per centrinį koordinuojantį modulį, negali kreiptis į lokalų mazgą, o centrinis mazgas parodo į kurį mazgą kreipsis.

**17. Lygiagrečiosios DB sistemos, jų savybės. Pagrindinės lygiagrečiųjų DBVS architektūros.**

Lygiagrečiose užklausa specialiai išskirstoma į smulkesnes ir apdorojama skirtinguose servuose skirtingais procesoriais, kad būtų efektyviau.

SAVYBES:geresnis kainos/našumo santykis, didesnis našumas, preciziškas, tikslingas plėtimas.

ARCHITEKTŪROS: su bendra atmintim – atmintis buferizuoja duomenis iš diskų, naudojama bendra magirstralė, kuri leidžia pasiekti globalią atminti iš skirtingų mazgų, o tie mazgai neturi atminties. Su bendrais diskais – mazgai kiekvienas turi savo procesorių, lokalią operatyvią atminti, diskai tarpusavy gali bendrauti ir pasiimti duomenis iš diskų ar gretimo mazgo atminties. Išskirstytais ištekliais - jokių bendrų išteklių, bendravimas pranešimais. Hibridinės archiektūros: klasteris, kuriame veikia du serverio mazgai ir klasteriai gali būt ivienas užklausas, kitas atsarginis. Nevienodos atminties – nebe atmintis, o sparčiai veikianti atmintis yra tarpusavyje sujungta ir magistralė kuri apjungia atminties blokus.

**18. Duomenų fragmentacija ir jos schema.**

**Duomenų dislokacijos schema.**

**Horizontalioji, vertikalioji, mišrioji fragmentacija, savybės.**

**Duomenų replikacija ir jos schema.**

**Replikacijos savybės.**

Aprašo, kokiu būdu turėtų būti gaunams bendras DB vaizdas iš atskirų fragmentų.

DISLOKACIJOS: nusako, kokiuose tinklo mazguose yra išdėstyti minimi DB fragmentai.

HORIZONTALIOJI: patogu išskirstyti į skirtingus mazgus, atskiri DB fragmentai išlaiko loginę bazės struktūrą.

VERTIKALIOJI: dalis lentelės atributų vienoj vietoje, kita dalis kitoj. Abiejuose fragmentuose tas pats raktas turi kartotis.

MIŠRI: naudojant abi fragmentacijas, išlaikytinos savybes: pilnumas, atstatomumas,nesikertamumas.

SAVYBES: efektyvu,geresnis našumas, saugumas – nepriimtina prieigos sparta,maža duomenų sauga.

REPLIKACIJA : duyomenų atkartojimas kitame serveryje. Patikimumas ir preinamumas labai išauga, tačiau darbas lėtas, sudėtingi konkurencinės kontrolės mechanizmai. Replikacijos schema – kiek, kur, kokių fragmentų kopijų daroma.

**19. Reliacinių duomenų modelių ir reliacinių duomenų bazių valdymo sistemų trūkumai.**

Nepakankamai gerai atvaizduojami realaus pasaulio objektai, silpnai palaikomi vientisumo ir korporatyviniai ribojimai, vienalytė duomenų struktūra, ribotas galimų operacijų rinkinys, rekursinio apdorojimo nebuvimas.

**20. Objektinės ir objektinės-reliacinės duomenų bazės ir jų samprata. Pagrindinės sąvokos:**

**inkapsuliacija, objektai, jų identifikacija, klasės, paveldėjimas, sudėtiniai objektai. Būtinos**

**objektinių DBVS savybės..**

Abi turi panašumų, skirtumas - objektiškai orientuotose DB reliacinis modelis išplėstas, stengiantis suteikti galimybę reliaciniu DB pagrindu kurti DB sistemas.

Objektinėms DB lengviau aprašyti sudėtingus modelius, naudojant objektines programavimo kalbas apdorojimui - mažesnės nesuderinamumo problemos, trūkumas - nėra griežto matematinio modelio, sunku standartizuoti.

SAVOKOS: inkapsuliacija – objekto savybes ir metodai nematomi iš išorės. Paveldėjimas - paveldi savo klasės ir aukštesnių klasių savybes. Klasės - objektas priklauso kuriai nors klasei, Sudėtiniai objektai – galima matyti jį kaip vientisą objektą, bet jis susideda iš smulkesnių, struktūra neaiški (objektas turi kitus objektus savyje).

**21. Objektinių duomenų bazių valdymo sistemų privalumai ir trūkumai. Permanentinės**

**programavimo kalbos.**

Našesnis programuotojo darbas, permanentiškumo užtikrinimo mechanizmus galima centralizuoti, programuotojas daugiau dėmesio skirs verslo f-joms. -gaištamas laikas automatiniam suderinamumui,neįmanoma pasiekti 100proc skaidrumą.

PERMANENTINĖS PROGRAMAVIMO KALBOS: kalbos, kuriomis apdorojami tarpiniai duomenys gali būti išsaugoti, o vėliau panaudoti kitų programų.DB programavimo kalbos yra permanentinės (SQL).

**22. Aktyviosios duomenų bazės, jų samprata, savybės. Pagrindiniai DB sistemų aktyvavimo**

**būdai, jų savybės. ECA taisyklės, jų samprata. Verslo taisyklės, trigeriai.**

Aktyvios DB gali įspėti taikomąsias programas, ekspoetrtines sistemas. Taisyklių generuojamais atnaujinimais atnaujinti pačią DB.

Tai sistema, gebanti reaguoti į aplinkos įvykius.

PAGRINDINIAI DB SISTEMŲ AKTYVAVIMO BŪDAI:

-Aktyvuoti DBVS;-Įdiegti DBVS aktyvumo savybes.

AKTYVUMO SAVYBES: decentralizuotas, t.y. kiekviena taikomoji programa turi savo kodo dalį, užtikrinančią DBVS aktyvumą, todėl gali kilti įvairių palaikymo problemų.

ECA TAISYKLĖS: įvykis, salyga, veiksmas (kai pasirodo įvykis, tikrinama salyga, jei ji tenkinama – vykdomas veiksmas)

VERSLO TAISYKLĖS, TRIGERIAI:taisyklių realizacijai naudojami trigeriai, tai yra aktyvūs DBVS elementai, kurie aktyvuojasi susidarius tam tirkoms sąlygoms (tvarko išvestinius duomenis, jų saugumą, kaitą, priežiūra ir pan.)

**23. Įvykiai, jų savybės, šaltiniai, apdorojimo problemos. Sąlygos, veiksmai. Įvykio-sąlygos bei**

**sąlygos-veiksmo jungimo būdai. Sąveika tarp taisyklių. Vykdymo baigtinumas ir jo patikra.**

Įvykis yra atsitikimas, įvykęs tam tikru laiko momentu. Šaltiniai: duomenų keitimas, elgsenos iškvietimas,transakcija, prieštaravimai,laikrodis, išorė.

Įvykių tipai: elementarieji, sudėtiniai.

ĮVYKIO-SALYGOS:apibrėžia kada, atsižvelgiant į taisyklę sužadinusį įvykį, turi būti įvertinta sąlyga.

SĄLYGOS-VEIKSMO:nurodo, kada turi būti vykdomas veiksmas, atsižvelgiant į sąlygos įvertinimą.

SUJUNGIMO BŪDAI: nedelsiamas,atidėtas,savarankiškas.

SĄVEIKA TARP TAISYKLIŲ: taisyklės vykdymas gali tiesiogiai arba netiesiogiai iššaukti kitą taisyklę, tada galimi du būdai: rekursinis arba iteracinis.

VYKDYMO BAIGTINUMAS IR JO PATIKRA: baigimasis yra esminė ir labiausiai pageidaujama savybė, reikėtu vengti situacijų, kai vartotojų sužadintos transakcijos sukelia neribotą vykdymą, nes viršijamas maksimalus rekursiniu būdu vykdomų taisyklių skaičius. Taisyklių baidtinumas dažniausiai tikrinamas aktyvavimo grafu.

1. Lygiagrečios DBVS, jų samprata ir pagrindinės architektūros.

2. Kuo skiriasi žvaigždės ir snaigės duomenų architektūros multidimensinėje DB.

3. Kaip suprantamas GUI standartas programų sistemoje.

4. Kaip suprantamas daugybinis objektų paveldimumas ir kuo jis problematiškas.

5. Kokius įrankius siūlo Microsoft, ruošiant ataskaitas SQL Server Reporting services

|  |  |  |
| --- | --- | --- |
| Paveikslėlis, kuriame yra žinutė  Automatiškai sugeneruotas aprašymas |  | **Kamilė Bikaitė | VGTU SA FMF** Vicepirmininkė El. paštas: [vice@fmf.vgtusa.lt](mailto:vice@fmf.vgtusa.lt) Tel. Nr.: +37062249631 Saulėtekio al. 11, SRL-I 504A,  [www.vgtusa.lt](http://www.vgtusa.lt/) |