

Samarqand davlat universitetning Kattaqo'rg'on filiali Aniq va tabiiy fanlar fakulteti 3-kurs Axborot tizimlari va texnologiyalari (tarmoqlar va sohalar bo'yicha) yo'nalishi Berilganlar bazasi fanidan yakuniy nazorat savollar

Imtihon shakli: Amaliy

№	Mavzu	1-QISM “OSON SAVOLLAR” deb nomalanadi va semestrda o'qitilishi rejalashtirilgan mavzularning kirish qismidan asosan nazariy savollardan iborat bo'ladi (tayanch iborasi bo'ladi).	2-QISM “ODDIY SAVOLLAR” deb nomalanadi va semestrda o'qitilishi rejalashtirilgan mavzularning 1- rejasidan asosan mulohazaviy savollardan iborat bo'ladi (tayanch iborasi bo'ladi).	3-QISM “MURAKKAB SAVOLLAR” deb nomalanadi va semestrda o'qitilishi rejalashtirilgan mavzularning 2- rejasidan asosan qoidalar, teoremlar, kabi savollardan iborat bo'ladi (tayanch iborasi bo'lmaydi).	4-QISM qism “MURAKKAB SAVOLLAR 2” deb nomalanadi va semestrda o'qitilishi rejalashtirilgan mavzularning 3- rejasidan asosan tendensiyalar, paradigmalar, konsepsiyalar kabi savollardan iborat bo'ladi (tayanch iborasi bo'lmaydi).	5-qism “QIYIN SAVOLLAR” deb nomalanadi va semestrda o'qitilishi rejalashtirilgan mavzularning xulosa yani yakuniy qismidan asosan talabani mustaqil fikrlashga, mavzuni to'liq anglashga yo'naltiradigan savollardan iborat bo'ladi (tayanch iborasi bo'lmaydi).
1.	Berilganlar bazasining maqsadi, vazifalari va asosiy tushunchalari	MySQL'da “university_db” nomli yangi ma'lumotlar bazasini yarating va “students” jadvalini loyihalang (id, name, age, major, enrollment_date). 	“students” jadvalida “major” bo'yicha guruhlangan talabalarni hisoblash uchun GROUP BY bilan ikki qatlamli so'rov yozing. 	“students” jadvalida “enrollment_date” qo'shib, so'nggi 6 oy ichidagi yangi talabalarni subquery bilan aniqlang. 	“students” jadvalidan “major” bo'yicha eng ko'p talaba bor bo'limni aniqlovchi subquery va JOIN bilan murakkab so'rov yozing. 	“students” jadvalida tranzaksiya yordamida 3 ta talaba ma'lumotini o'chirib, “students_archive” jadvaliga ko'chirib saqlang.
2.		“students” jadvaliga 5	“age” va “major”	“students” jadvalida	“students” jadvalida	Ma'lumotlar bazasining

		ta turli mutaxassislikdagi talaba qo'shing (masalan, major: 'CS', 'EE').	bo'yicha filtrlovchi murakkab WHERE shartini qo'shing (age > 20 AND major LIKE '%CS%').	“total_credits” uchun default (30) va cheklov (credits > 0) qo'shing.	“gpa” (FLOAT) qo'shib, 3.5 dan yuqori GPA'li talabalarni tahlil qiling.	maqsadini hisobga olib, “university_db” uchun ER-diagrammni murakkab tarzda loyihalang va optimallashtirish taklif qiling.
3.	Berilganlar bazasining arxitekturasi va uch bosqichli arxitektura	“company_db” nomli yangi ma'lumotlar bazasini yarating va “departments” jadvalini loyihalang (id, dept_name, location). 	“departments” va “employees” (id, name, dept_id) jadvallarini JOIN qilib, har bir bo'limdagi xodimlar sonini hisoblash uchun so'rov yozing. 	“employees” jadvalida “salary” va “dept_id” bo'yicha umumiy xarajatni hisoblash uchun agregat funksiyalar va GROUP BY ishlatng. 	“employees” va “departments” o'rtasidagi munosabatlarni optimallashtirish uchun “salary” va “dept_id” uchun kompozit indeks yarating. 	Uch bosqichli arxitekturada “employees” uchun foydalanuvchi interfeysi uchun murakkab ko'rinish (view) yarating va xavfsizlik siyosatini sinovdan o'tkazing.
4.		“departments” jadvaliga 4 ta bo'lim qo'shing (masalan: IT, HR, Finance, Sales).	Uch bosqichli arxitekturada “employees” uchun xususiy ko'rinish (view) yarating va shartli filtrlash qo'shing.	“departments” da “manager_id” uchun self-join bilan boshqaruv zanjirini aniqlovchi murakkab so'rov yozing.	Uch bosqichli arxitekturada xavfsizlikni ta'minlash uchun “employees” uchun trigger loyihalang.	“company_db” uchun ma'lumotlar bazasi arxitekturasini optimallashtirish uchun umumiy strategiya taklif qiling.
5.	Berilganlar bazasi modellari. Infologik model. ER-model	“school_db” nomli yangi ma'lumotlar bazasini yarating va “classes” (id, class_name) jadvalini loyihalang. 	“students” (id, name) va “classes” o'rtasida “enrollments” (student_id, class_id) jadvalini loyihalang va foreign key qo'shing. 	“enrollments” da har bir talabaning o'tagan kurslar sonini hisoblash uchun subquery va COUNT ishlatng. 	“enrollments” da cheklovlar bilan birga trigger yarating (masalan, kursga yozilishda maksimal limit 50). 	Infologik model asosida “school_db” uchun murakkab ER-diagrammni loyihalang va tranzaksiyaviy birdamlikni ta'minlang.
6.		“classes” jadvaliga 3 ta kurs qo'shing (masalan: Math, Physics, History).	“enrollments” da 5 ta qator qo'shib, har bir talabaning kurslarini ko'rsatuvchi JOIN so'rovini yozing.	ER-modelda “teachers” (id, name) jadvalini qo'shib, “classes” bilan “1:N” munosabatini amalga oshiring.	“students”, “classes” va “teachers” o'rtasidagi munosabatlarni optimallashtirish uchun indekslar qo'shing.	“enrollments” jadvalida kurslar bo'yicha tahlil qiluvchi murakkab protsedura yarating.
7.	Berilganlar bazasini boshqarish	“system_db” nomli yangi ma'lumotlar bazasini yarating va	“users” jadvalida “role” bo'yicha foydalanuvchilarni	“users” uchun “admin” rollariga GRANT va REVOKE huquqlarini	“users” uchun xususiy ko'rinish (view) yarating va faqat “admin” ga	“users” jadvali asosida MBBT'da foydalanuvchi

	tizimlari (MBBT)	“users” (id, username, role) jadvalini loyihalang. 	guruhlab, har bir rolning sonini hisoblash uchun GROUP BY ishlatng. 	boshqaruvchi murakkab so‘rovni yozing. 	ochiq bo‘lsin, qolganlarga cheklov qo‘ying. 	huquqlarini murakkab tarzda boshqaruvchi protsedura yarating.
8.		“users” jadvaliga 4 ta foydalanuvchi qo‘shing (role: admin, user).	“users” da “last_access” (TIMESTAMP) qo‘shib, oxirgi 30 kun ichidagi faol foydalanuvchilarni aniqlang.	“users” da “password” ni SHA-256 bilan shifrlash uchun UPDATE so‘rovini loyihalang.	“users” da tranzaksiya yordamida bir nechta foydalanuvchi ma'lumotlarini o‘zgartirib, xavfsizlikni sinab ko‘ring.	“system_db” uchun xavfsizlik siyosatini optimallashtirish uchun umumiy strategiya taklif qiling.
9.	Berilganlar bazasini rejalashtirish, loyihalash va administratorlash	“project_db” nomli yangi ma'lumotlar bazasini yarating va “projects” (id, name, budget) jadvalini loyihalang. 	“projects” da “budget” bo‘yicha eng qimmat 3 loyihani aniqlovchi TOP-N so‘rovini yozing. 	“projects” da “budget” dan oshib ketgan loyihalarni aniqlash uchun trigger yarating. 	“projects” jadvalida “budget” va “start_date” uchun kompozit indeks yarating va samaradorligini tekshiring.	“projects” jadvali asosida administratorlik uchun murakkab protsedura yarating va xavfsizlikni ta'minlash strategiyasini taklif qiling.
10.		“projects” jadvaliga 4 ta loyiha qo‘shing (budget: 5000-20000).	“projects” ga “start_date” va “end_date” (DATE) qo‘shib, davomiyligni hisoblash uchun DATEDIFF ishlatng.	“projects” uchun “project_manager” (INT) qo‘shib, “managers” jadvaliga foreign key bilan bog‘lang.	“projects” da tranzaksiya yordamida bir nechta loyiha byudjetini o‘zgartirib, xavfsizlikni sinab ko‘ring.	“project_db” uchun ma'lumotlar bazasini rejalashtirish va optimallashtirish uchun umumiy loyiha taklif qiling.
11.	Berilganlar bazasida relyatsion model	“sales_db” nomli yangi ma'lumotlar bazasini yarating va “customers” (id, name) jadvalini loyihalang.	“customers” va “orders” (id, customer_id, amount) jadvallarini “1:N” munosabati bilan JOIN qilib so‘rov yozing.	Har bir mijozning umumiy buyurtma summasini hisoblash uchun subquery va SUM ishlatng.	“orders” ga “order_status” (ENUM) qo‘shib, statusga qarab filtrlovchi murakkab so‘rovni yozing.	Relyatsion model asosida “sales_db” uchun murakkab munosabatlar tizimini loyihalang va tranzaksiyaviy birdamlikni ta'minlang.
12.	va munosabatlar	“customers” jadvaliga 5 ta mijoz qo‘shing.	“orders” da “customer_id” uchun foreign key cheklovini qo‘shing.	“orders” va “customers” o‘rtasidagi munosabatlarni optimallashtirish uchun indekslar qo‘shing.	“deliveries” jadvalini qo‘shib, “orders” bilan “1:1” munosabatini loyihalang.	“customers”, “orders” va “deliveries” o‘rtasidagi tahlil qiluvchi protsedura yarating.
13.	Relyatsion algebra va relyatsion	“store_db” nomli yangi ma'lumotlar bazasini yarating va “products”	“products” da “price” bo‘yicha 100 dan katta mahsulotlarni saralash	“products” dan eng kam stockdagi 3 mahsulotni aniqlovchi TOP-N	“products” da “category” qo‘shib, kategoriyaga qarab MAX va MIN	Relyatsion hisoblash asosida “products” uchun murakkab tahlil

	hisoblash elementlari	(id, name, price) jadvalini loyihalang.	uchun so‘rov yozing.	so‘rovni yozing.	qiymatlarni topuvchi murakkab so‘rovni yozing.	qiluvchi protsedura yarating.
14.		“products” jadvaliga 5 ta mahsulot qo‘shing (price: 50-300).	“price” va “stock” bo‘yicha guruhlash uchun GROUP BY ishlatng.	“price” va “stock” asosida umumiy qiymatni hisoblash uchun agregat funksiyalarni birlashtiring.	“products” va “sales” jadvallarini JOIN qilib, sotuvlar bo‘yicha tahlil qiling.	“store_db” uchun so‘rov samaradorligini oshirish uchun umumiy optimallashtirish strategiyasi taklif qiling.
15.	Berilganlar bazasini normallashtirish	“trade_db” nomli yangi ma'lumotlar bazasini yarating va “sales_data” (id, customer, product, amount) jadvalini loyihalang.	“sales_data” da takrorlanishlarni aniqlash uchun DISTINCT bilan so‘rov yozing.	“customers” va “products” jadvallarida primary key va foreign key qo‘shib, 2NF ga olib boring.	Normallashtirish natijasini tahlil qilib, “sales_data” jadvalining samaradorligini oshirish uchun indekslar qo‘shing.	“trade_db” uchun normallashtirish jarayonini murakkab tarzda qayta loyihalang va tranzaksiyaviy birdamlikni ta'minlang.
16.		“sales_data” jadvaliga 5 ta qator qo‘shing.	“sales_data” ni “customers” va “products” jadvallariga ajratishni rejalang.	“sales_data” ni 3NF ga olib borish uchun funksional qaramliklarni aniqlang.	“sales_data” ni 3NF dan keyin so‘rov optimallashtirish strategiyasi taklif qiling.	Normallashtirish natijasida umumiy ma'lumotlar bazasi tuzilmasini optimallashtiruvchi loyiha yarating.
17.	Ma'lumotlar bazasida normalizatsiya (1NF, 2NF, 3NF va kodd)	“order_db” nomli yangi ma'lumotlar bazasini yarating va “order_items” (order_id, product, qty, price) jadvalini loyihalang.	“order_items” ni 1NF ga moslashtirish uchun takrorlanishlarni tekshiring.	“order_items” da funksional qaramliklarni aniqlab, 3NF ga olib boring.	Normalizatsiyadan keyin “order_items” da indekslar qo‘shib, so‘rov samaradorligini oshiring.	“order_db” uchun 3NF ga asoslangan murakkab tahlil qiluvchi protsedura yarating.
18.		“order_items” jadvaliga 5 ta qator qo‘shing.	“order_items” ni 2NF uchun “products” jadvaliga ajratishni rejalang.	“order_items” da tranzaksiyaviy birdamlik uchun trigger loyihalang.	“order_items” va “orders” o‘rtasidagi munosabatni ER-diagramda optimallashtirishni taklif qiling.	Normalizatsiya natijasida ma'lumotlar bazasini umumiy tarzda qayta loyihalang.
19.	SQL tili. SQL operatorlarini yozish	“hr_db” nomli yangi ma'lumotlar bazasini yarating va “employees” (id, name,	“employees” da “dept” bo‘yicha har bir bo‘limning xodimlar sonini hisoblash uchun	“employees” da “name” da “a” harfi bor xodimlarni REGEXP bilan aniqlovchi so‘rovni yozing.	Tranzaksiya yordamida “employees” da “salary” ni 10% oshiring va natijani tekshiring.	“employees” jadvalida murakkab tranzaksiyaviy o‘zgarishlar (INSERT,

		dept, salary) jadvalini loyihalang.	GROUP BY yozing.			UPDATE, DELETE) uchun protsedura yarating.
20.		“employees” jadvaliga 5 ta xodim qo‘shing.	“salary” da 5000 dan katta xodimlarni filtrlovchi WHERE shartini qo‘shing.	“employees” da “hire_date” qo‘shib, 2023 yilgacha ishlaganlarni toping.	“dept” bo‘yicha eng yuqori o‘rtacha maoshli bo‘limni aniqlovchi subquery yozing.	“hr_db” uchun SQL operatorlarini optimallashtirish uchun umumiy strategiya taklif qiling.
21.	SQLda bazaviy va foydalanuvchi ma'lumot turlari	“warehouse_db” nomli yangi ma'lumotlar bazasini yarating va “inventory” (id, item_name, price) jadvalini loyihalang.	“inventory” da “category” (ENUM) qo‘shib, kategoriyaga qarab mahsulotlarni guruhlashni amalga oshiring.	“price” uchun CHECK (price > 0) cheklovini qo‘shing. “inventory” ga “last_updated” (TIMESTAMP) qo‘shib, oxirgi 7 kun ichidagi o‘zgarishlarni aniqlovchi so‘rov yozing.	“inventory” da “supplier_id” qo‘shib, “suppliers” jadvaliga foreign key bilan bog‘lang.	“inventory” uchun foydalanuvchi tomonidan aniqlangan ma'lumot turi (user-defined type) qo‘shib, murakkab tahlil qiling.
22.		“inventory” jadvaliga 5 ta mahsulot qo‘shing.	“price” uchun CHECK (price > 0) cheklovini qo‘shing.	“item_name” uchun UNIQUE cheklovini qo‘shib, takrorlanishni olding.	“inventory” da indekslar yordamida so‘rov samaradorligini oshiring.	“warehouse_db” uchun ma'lumot turlarini optimallashtirish strategiyasi taklif qiling.
23.	Tranzaksiyalarni boshqarish va so‘rovlar yaratish va qayta ishlash	“bank_db” nomli yangi ma'lumotlar bazasini yarating va “accounts” (id, balance) jadvalini loyihalang.	“accounts” da “balance” dan 1000 dan katta hisoblarni filtrlovchi so‘rov yozing.	Tranzaksiya yordamida ikki hisob o‘rtasida 500 so‘m pul o‘tkazing.	“accounts” da “transaction_log” jadvalini yarating va tranzaksiyalarni yozuvchi trigger qo‘shing.	“accounts” uchun murakkab tranzaksiyaviy boshqaruv protsedurasi yarating va xavfsizlikni ta'minlang.
24.		“accounts” jadvaliga 4 ta hisob qo‘shing (balance: 1000-5000).	“accounts” ga “last_transaction” (DATE) qo‘shing.	Tranzaksiya muvaffaqiyatsiz bo‘lsa, ROLLBACK bilan qaytaring.	“accounts” da bir nechta tranzaksiyani birlashtiruvchi protsedura loyihalang.	“bank_db” uchun tranzaksiya optimallashtirish strategiyasi taklif qiling.
25.	XML va ma'lumotlar bazasi	“xml_db” nomli yangi ma'lumotlar bazasini yarating va “xml_records” (id, xml_data TEXT) jadvalini loyihalang.	“xml_records” da XML ichidagi “id” qiymatini EXTRACTVALUE bilan aniqlovchi so‘rov yozing.	“xml_records” da XML ma'lumotlarini jadvalga aylantiruvchi murakkab so‘rovni yozing. 	“xml_records” da XML validatsiyasi uchun trigger yarating.	“xml_records” dan XML ma'lumotlarini ajratib, murakkab tahlil qiluvchi protsedura yarating.

26.		“xml_records” ga 3 ta murakkab XML qo‘shing.	XML dan “name” va “age” qiymatlarini ajrating.	XML ni yangilab, yangi element (discount) qo‘shing.	XML ma'lumotlarini tahlil qiluvchi ko‘rinish (view) loyihalang.	“xml_db” uchun XML va SQL sinxronizatsiyasini optimallashtiruvchi strategiya taklif qiling.
27.	JSON Ma'lumotlar Strukturasining Tahlili	“json_db” nomli yangi ma'lumotlar bazasini yarating va “json_store” (id, json_data JSON) jadvalini loyihalang.	“json_store” da JSON ichidagi “product” qiymatini JSON_EXTRACT bilan aniqlovchi so‘rov yozing.	“json_store” da JSON ichidagi “category” bo‘yicha guruhlash qiling.	“json_store” da JSON ma'lumotlarini tahlil qiluvchi ko‘rinish (view) yarating.	“json_store” uchun murakkab tahlil qiluvchi protsedura yarating.
28.		“json_store” ga 3 ta murakkab JSON qo‘shing.	JSON dan “price” va “stock” qiymatlarini filtrlang.	JSON ni yangilab, “discount” maydoni qo‘shing (JSON_SET)	JSON va jadval ma'lumotlari o‘rtasidagi konvertatsiyani optimallashtiruvchi so‘rov yozing.	“json_db” uchun JSON ma'lumotlar tuzilmasini umumiy tarzda optimallashtiruvchi strategiya taklif qiling.
29.	Ma'lumotlar bazasini administratorlash va xavfsizligini ta'minlash	“security_db” nomli yangi ma'lumotlar bazasini yarating va “security_users” (id, username) jadvalini loyihalang.	“security_users” da “password” ni SHA-256 bilan shifrlash uchun UPDATE so‘rovini yozing.	“security_users” da “access_level” (INT) qo‘shib, GRANT huquqlarini boshqaruvchi so‘rovni yozing.	“security_users” uchun xususiy ko‘rinish yarating va faqat “admin” ga ruxsat bering.	“security_db” uchun murakkab xavfsizlik boshqaruv protsedurasi yarating.
30.		“security_users” ga 4 ta foydalanuvchi qo‘shing.	“username” bo‘yicha foydalanuvchilarni saralovchi so‘rov yozing.	“login_attempts” jadvalini qo‘shib, trigger bilan muvaffaqiyatsiz kirishlarni kuzating.	“security_users” da tranzaksiya yordamida xavfsizlikni sinovdan o‘tkazing.	“security_db” uchun xavfsizlikni umumiy tarzda optimallashtiruvchi strategiya taklif qiling.