

1es típus (tranzakciós tulajdonságok)

Párosítsa össze az egyes tranzakció tulajdonságokat és leírásukat!

Durability	A tranzakció sikeres lefutása után a változás tartósan megmarad	✓
Isolation	A párhuzamosan futó tranzakciók nem zavarhatják egymást	✓
Consistency	A tranzakció végrehajtása után az állapot konzisztens marad	✓
Atomicity	A tranzakció nem valósulhat meg részlegesen	✓

Atomicity A tranzakció nem valósulhat meg részlegesen

Consistency A tranz. végrehajtása után az állapot konzisztens marad (pl. kényszerek teljesülnek)

Isolation A párhuzamosan futó tranzakciók nem zavarhatják egymást

Durability Tranz. sikeres lefutása után a változás tartósan megmarad

2es típus (tranzakciós módok)

Válassza ki a hiányzó szavakat!

Az explicit tranzakciókat mi magunk definiáljuk a(z) parancs segítségével. Az explicit tranzakciók egymásba is ágyazhatók. Ilyenkor a(z) változó mondja meg, hogy hányadik szinten vagyunk. Minden BEGIN TRANSACTION 1-gyel növeli, minden 1-gyel csökkenti a változó értékét

Explicit tranz-kat mi magunk definiáljuk a **BEGIN TRANSACTION** paranccsal

Az explicit tranzakciók egymásba is ágyazhatók. Ilyenkor a **@@TRANCOUNT** változó mondja meg, hogy hányadik szinten vagyunk

((Kezdetben, illetve ROLLBACK után a **@@TRANCOUNT** értéke 0))

Minden **BEGIN TRANSACTION** 1-gyel növeli, minden **COMMIT** 1-gyel csökkenti a **@@TRANCOUNT** értékét

3as típus (Tárolt eljárások)

Húzza be a megfelelő válaszokat a szövegbe!

Tárolt eljárások használata esetén a kliens-szerver üzenetek száma. A tárolt eljárásokat egyszerre használhatja. A tárolt eljárások az adatbázisban találhatók.

CSÖKKEN a kliens-szerver üzenetek száma

egyszerre **TÖBB ALKALMAZÁS IS** használhatja

az adatbázisban **KÖZPONTI HELYEN**

4es típus

A tárolt eljárások a SELECT utasításban is használhatók.

Válasszon ki egyet:

☒ Igaz

☐ Hamis

HAMIS

5ös típus

Ha A funkcionálisan meghatározza B-t, akkor azt is mondhatjuk, hogy B funkcionálisan függ A-tól

Válasszon ki egyet:

☐ Igaz

☒ Hamis

IGAZ

6os típus

Válassza ki az adatmodell építőelemeit!

Válasszon ki egyet vagy többet:

☐ a. Rekord

☒ b. Egyed ✓

☒ c. Tulajdonság ✓

☐ d. Kapcsolat

Adatmodell építőelemei:

EGYED

TULAJDONSÁG

KAPCSOLAT

Melyik relációs algebrai művelethez tartozik a következő leírás?

A művelet során a reláció oszlopaiból csak bizonyosakat tartunk meg adott sorrendben.

Válasszon ki egyet:

☒ a. Projekció ✓

☐ b. Természetes összekapcsolás

☐ c. Metszet

☐ d. Szelekció

A **projekció** művelete során a reláció oszlopaiból csak bizonyosakat tartunk meg adott sorrendben.

A **szelekció** művelete során a reláció soraiból csak az adott feltételnek megfelelőket tartjuk meg.

Az **egyesítés** (unió) művelet során két azonos szerkezetű reláció sorainak unióját kapjuk meg.

A **metszet** művelet során két azonos szerkezetű reláció közös sorait kapjuk meg.

Két azonos szerkezetű reláció **különbségének** eredménye azokat a sorokat tartalmazza, amelyek az első relációban benne vannak, de a másodikban nincsenek.

A **Descartes-szorzat** művelet két relációs sorait teszi egymás mellé minden lehetséges kombinációban.

A **természetes összekapcsolás** művelete két relációt kapcsol össze egy-egy attribútumérték összehasonlításával.

12es (tranzakció fogalma)

Válassza ki a hiányzó szavakat!

A tranzakció utasítások olyan sorozata, amelyet egyetlen logikai egységként kezelhetünk. A tranzakció végén vagy minden változást érvényesítünk , vagy minden egyes lépést visszavonunk .

13as (trigger, view)

Húzza be a válaszokat a megfelelő helyekre!

A(z) olyan speciális eljárások, amelyek DML utasítás előtt, után vagy helyett futnak le. A(z) egy elmentett, névvel ellátott lekérdezés. A lekérdezések sebességét gyorsítani lehet, ha az adatokat . A(z) a táblához vagy nézethez rendelt olyan speciális adatstruktúra, amely felgyorsítja a lekérdezések sebességét.

14es (**B-fák tulajdonságai**)

B-fák esetén a gyökértől a levelekig vezető utak hossza egyforma.

Válasszon ki egyet:

☒ Igaz

☐ Hamis

A gyökértől a levelekig vezető utak hossza egyforma

Az indexek a B-fa csomópontjaiban helyezkednek el

Az adatok helyét jelző mutató csak a levelekben található

A struktúra lehetővé teszi a soros és a random elérést is

15ös (indexek) *25ös clustered

Párosítsa össze az alábbi index adatstruktúrákat és jellemzőiket:

Hash-alapú index	Az adatok csoportba vannak rendezve. Egy függvény adja meg, hogy melyik csoportban van az adat.
Többosztású index	Választás...
Bitmap index	Választás...
Egyszintű index	Az adatok csoportba vannak rendezve. Egy függvény adja meg, hogy melyik csoportban van az adat. Két mezőből álló indextábla, amely az indexelt mező alapján sorba van rendezve. Olyan oszlopokra alkalmazzuk, ahol kevés az egyedi érték.

Az indexekhez is indexet készítünk.

HASH – adatok csoportokba vannak rendezve, 1 fgv adja meg hogy melyik csop-ban van az adat

BITMAP – olyan oszlopokra alkalmazzuk ahol kevés az egyedi érték

EGYSZINTES – 2 mezőből álló indextábla amely az indexelt mező alapján sorba van rendezve

TÖBBSZINTES – az indexekhez is indexet készítünk

CLUSTERED – Az adatokat az index kulcsnak megfelelő sorrendbe rendezi és tárolja.

NON-CLUSTERED – A non-clustered index kulcs-mutató érték párokat tárol.

COLUMNSTORE – A columnstore index jellemzője az oszlop-alapú tárolás és lekérdezés végrehajtás

SÜRŰ – Az indexmutató egy rekordra mutat.

RITKA – Az indexmutató egy blokkra mutat. A blokkon belül a keresés szekvenciális

16os (elkülönítési -izolációs- szintek)

Melyik a tranzakció izolációs szintre jellemző, hogy a nem véglegesített adatok is olvashatók?

Válasszon ki egyet:

- ☐ a. Repeatable read
- ☐ b. Read committed
- ☐ c. Serializable
- ☒ d. Read uncommitted ✓

Read uncommitted: minden adat olvasható (a nem véglegesítettek is)

Read committed: csak a véglegesített (COMMITTED) adatok olvashatók (alapértelmezett szint)

Repeatable read: az olvasott adatot nem módosíthatja más tranzakció

Serializable: az olvasott adathalmazra nem engedélyezett az új adat beszúrása sem

17es (felh.-i fgy-ek VS tárolt eljárások) *24es folytatás tárolt eljárások

Melyik állítás NEM igaz az SQL-ben létrehozott felhasználói függvényekre?

Válasszon ki egyet:

- ☒ a. Lehetnek input paraméterek ✗
- ☐ b. Lehetnek saját változói
- ☒ c. Tranzakciók is használhatók ✓
- ☐ d. A SELECT utasításban használhatók

Függvények	Tárolt eljárások
Csak input paraméterek	Input és output paraméterek
Tranzakciók nem használhatók	Tranzakciók is használhatók
A SELECT utasításban használhatók	A SELECT utasításban nem használhatók
Kivételkezelés nem használható	Kivételkezelés használható
Nem hívhat meg tárolt eljárást	Függvényhívás lehetséges
Mindig egy értéket ad vissza	Visszaadhat nulla, egy vagy több értéket

18as (jogosultságok)

Párosítsa össze a jogosultságokkal kapcsolatos fogalmak leírását és a hozzájuk tartozó fogalmakat!

Azok az felhasználói fiókok, amelyekkel a felhasználók hozzáférhetnek az SQL-szerverhez	LOGIN ✓
Azok a rekordok, amelyek az SQL-szerveren kívüli erőforrásokhoz való csatlakozáshoz szükséges hitelesítési információkat tartalmazzák	SECURITY PRINCIPAL ✗
Azok az identitások, akik számára jogosultságok megadhatók	CREDENTIAL ✗

SECURITY PRINCIPAL -- identitások, akik számára jogosultságok megadhatók

LOGIN -- felh. fiókok amely hozzáfér az SQL-het

CREDENTIAL -- rekordok amelyek az sql-szerveren kívüli erőforrásokhoz való csatlakozáshoz az infót tartalmazzák

SECURABLE -- objektumok, amelyhez jogosultság rendelhető

19es (adatbázis -OLTP- vs adattárház)

Válassza ki a hiányzó szavakat!

Az OLTP rendszerekre jellemző a(z) [magas fokú] ✗ normalizáltság, és a felhasználók magas száma. Az egyidejű tranzakciók gyakoriak ✓. A lekérdezések többnyire [elemi] ✗ adatokat használnak fel.

- ☐ Sok felhasználó
 - ☐ Sok kicsi, konkurens tranzakció
 - ☐ Rengeteg SQL utasítás (pl. másodpercenként több ezer, esetenként még ennél is több)
 - ☐ Az utasítások önmagukban leginkább egyszerűek
 - ☐ A lekérdezések döntő többsége kevés sort érint
 - ☐ A tranzakciókezelés a fő kihívás, az utasítások végrehajtása általában könnyű
 - ☐ Elvárt a magasfokú normalizáltság: minél inkább elkerüljük az anomáliákat
- ☐ Viszonylag kis számú felhasználó (nevezik őket adatelemzőknek, adatbányászoknak is)
 - ☐ Viszonylag kis számú, de gyakran igen nehéz lekérdezés
 - ☐ Nem jellemzők az egyidejű tranzakciók – nem akkora gond a konzisztencia
 - ☐ Gyakran nem jellemző az adatok online változtatása
 - ☐ A lekérdezések jellemzően összesített adatokat kérnek
 - ☐ Jellemzően nem cél az adatok normalizáltsága, hanem az ún. csillag séma (star schema) szerinti adatmodell

19es (adattavak vs adattárházak)

Data lake: Nagyvállalati szintű adatmenedzsment-platform, amelyen a különböző forrásokból származó adatok natív formátumukban érhetők el elemzésre.

- ☐ Big Data rendszerekre jellemző
- ☐ Központosított adatkezelés
- ☐ Az adattárházaknál rugalmasabb struktúra
- ☐ Strukturált és nem strukturált adatokat is tartalmazhatnak

Data lake	Data warehouse
Az adatok tárolásának célja előre nem definiált	Előre definiált tárolási cél
Az adatok nyers formában tárolódnak	Az adatok lekérdezésre alkalmas formában tárolódnak
Adattudósok, adatelemzők használják	Üzleti felhasználók használják
Feltörekvő technológia	Kidolgozott technológia
NoSQL lekérdezések	SQL lekérdezések
Gyors válaszdő	Lassú válaszdő
Alacsony költségű tárolás	Magas költségű tárolás

20as(adattárház modellek)

Húzza be a hiányzó szavakat a megfelelő helyekre!

Inmon adattárház modelljében a fókusz a(z) ☒ van. Az adattárház ☒ adatokat tartalmaz normalizált formában Az adatpiacok összegzett adatokat tárolnak ☒ , dimenzionális modellben

A fókusz az adattárházon (DW) van
 Az adattárház elemi adatokat tartalmaz normalizált formában
 Az adatpiacok összegzett adatokat tárolnak téma specifikus, dimenzionális modellben
 Az architektúra fontosabb rétegei a staging area, a DW, és az adatpiacok
 A felhasználó lekérdezhetnek akár az adatpiacokból, akár az adattárházból is

IMMON ADATTÁRZÁS MODELL

- A fókusz az adatpiacokon van
- Az adatpiacok tartalmaznak elemi és összegzett adatokat is
- Az adatpiacok csillag szerkezetűek
- Az architektúra legfontosabb részei a stage terület és az adatpiacok

KIMBALL MODELL

Kimball modellje a megfelelő, ha	Inmon modellje a megfelelő, ha
<ul style="list-style-type: none"> • a felhasználók IT területéről kerülnek ki • inkább taktikai döntések szükségesek • a forrásrendszerek viszonylag stabilak • minél előbbi eredményt szeretnénk elérni, kis kezdeti befektetéssel és csapattal • az adatok különálló üzleti területekről jönnek • a változások köre limitált 	<ul style="list-style-type: none"> • a felhasználók nem IT szakemberek • stratégiai döntések vannak túlsúlyban • a forrásrendszerek gyakran változnak • több idő, pénz és nagyobb létszámú csapat áll rendelkezésre • vállalati szintű adatintegráció szükséges • a változások köre bővíthet.

21 es (adattárház projektek → azok jellemzői)

<p>Adattárház projekteknel a SCRUM-módszertan általában jól használható.</p> <p>Válasszon ki egyet:</p> <p><input checked="" type="radio"/> Igaz <input checked="" type="checkbox"/></p> <p><input type="radio"/> Hamis</p>	<p>Az alábbiak közül melyek igazak az adattárház projektekre?</p> <p>Válasszon ki egyet vagy többet:</p> <p><input checked="" type="checkbox"/> a. A szokásos tesztelési módszerek nem mindig sikeresek</p> <p><input checked="" type="checkbox"/> b. A használati esetek száma többnyire magas</p> <p><input type="checkbox"/> c. A Scrum módszertan jól használható</p> <p><input type="checkbox"/> d. A megfelelő adatminőség biztosítása általában nem okoz gondot</p>
---	--

Adatok integrálása több forrásrendszerből

Sok (akár több száz) használati eset

A komplex működés miatt egyetlen felhasználó sem látja át teljesen a rendszert

Nehéz pontosan specifikálni a feladatot

A követelmények pontatlansága miatt sok hiba csak teszteléskor derül ki

A rendszertervhez képesti verifikálás nem elég

A felhasználó általi validáláson van a hangsúly

A szokásos tesztelési módszertanok nem mindig sikeresek

Az adattárház funkciók fejlesztési ciklusai hosszúak, számuk és időtartamuk nehezen tervezhető
Az egyes funkciókat nehéz szétválasztani, azok egymással rendszerint korrelálnak

A manapság divatos Scrum módszertan általában nem jól használható

Az IT és az üzlet másképpen gondolkodik
Szükség van motivációra a kulcsfelhasználók részéről is
A felsővezetés támogatása elengedhetetlen

A siker a Business Analyst és a megrendelő (kulcsfelhasználók) közötti kommunikáción múlik

Inkonzisztens adatok a forrásrendszerekben
Duplikációk
Hiányzó adatok
Logikai ellentmondások

Az egyik legnehezebb feladat az adatminőség biztosítása

24es(tárolt eljárások azok előnyei)

Húzza be a megfelelő válaszokat a szövegbe!

Tárolt eljárások használata esetén a kliens-szerver szimmetria. A tárolt eljárásokat egyre több alkalmazás is használhatja. A tárolt eljárások az adatbázisban találhatók.

központi helyen elhelyezkedik csak egy alkalmazás több helyen elhelyezkedik

Hatékonyaság	Fenntarthatóság	Biztonság	Üzleti logika elkülönítése
<ul style="list-style-type: none"> Egyszerre több alkalmazás is használhatja őket Csökken a szerver-kliens üzenetek száma 	<ul style="list-style-type: none"> A kódok egy központi helyen találhatók A módosítás, tesztelés elkülönülhet a tárolt eljárást hívó alkalmazástól 	<ul style="list-style-type: none"> Használatukkal korlátozható a táblákhoz való hozzáférés A hozzáférés biztosítása így nem a tárolt eljárást hívó alkalmazás feladata 	<ul style="list-style-type: none"> Az üzleti logika elkülönül a tárolt eljárást hívó alkalmazástól Csökkenhet a kliens programok miatti adathibák száma

25ös(clustered index)

Melyik állítás igaz a clustered indexre?

Válasszon ki egyet:

☐ a. Az adatok tárolási sorrendje nem felel meg az indexkulcs értékeinek

☐ b. Bitmap indexet használ

☐ c. Elsődleges kulcs létrehozásakor automatikusan létrejön

☐ d. Egy tábla esetén több is lehet belőle

VÁLASZ: C

Az adatokat az index kulcsnak megfelelő sorrendbe rendezi és tárolja.

- A clustered index B-fa struktúrát használ
- Egy tábla esetén csak egy clustered index hozható létre
- Alapértelmezés szerint az elsődleges kulcs definiálásakor automatikusan létrejön

27es(nézetek -view-)

Válassza ki a hiányzó szavakat!

Virtuális nézet esetén: csak a lekérdezés tárolódik, míg materializált nézet esetén: az adatok is tárolásra kerülnek.

A nézetek egyik előnye, hogy velük egyszerűbb a DML (DELETE, INSERT, UPDATE) műveletek elvégzése.

Nézetek előnye:

- csak a lekérdezés tárolódik
- gyorsabban futnak a lekérdezések
- DML műveletek
- korlátozható az adatok elérése
- DQL (SELECT)
- az adatok is tárolásra kerülnek

A Nézetek előnyei	
Korlátozható az adatok elérése	A bonyolultabb lekérdezések egyszerűbb formára hozhatók
Az adatokat többféle nézőpontból szemlélhetjük	Az adatfüggetlenség biztosítása

A Nézetek két fő típusa	
Virtuális	Materializált
<ul style="list-style-type: none"> Csak a lekérdezés tárolódik 	<ul style="list-style-type: none"> Az adatok is tárolásra kerülnek