# Feladatok – 2023.06.05

## I. rész

Definiálja következő fogalmakat!

* Relációséma
* Hivatkozási integritási megszorítás
* Második normálforma
* Equijoin
* Tranzakció elkülönítése (ACID)

## II. rész

1. Definiálja az Armstrong-axiómákat, és azok közül bizonyítsa be a reflexivitás szabályát!
2. Képezze le az alábbi ER diagramot a Neptun adatbázisról relációs modellé! Indokolja a lépéseit!

## III. rész

1. Osztályozza a kapcsolattípusokat (fogalom és definíció)!
2. Ismertesse a háromséma-architektúra szintjeit!
3. Ismertesse a relációalgebra szelekció műveletét és annak tulajdonságait!
4. Ismertesse a relációs adatbázis-tervezés nem hivatalos irányelveit!
5. Sorolja fel egy tranzakció lehetséges állapotait!
6. Mit jelent a tranziencia és perzisztencia az OO adatbázisokban?
7. Sorolja fel a NoSQL adatmodelleket!

# Megoldások – 2023.06.05

## I. rész

### Relációséma

Relációséma alatt az R(A1, A2, …, An) jelölést értjük, ahol R a relációsséma neve, A1, A2, …, An attribútumok. Minden Ai attribútum egy szerepkör neve, amelyet valamely Di tartomány játszik. Di-t az Ai attribútum tartományának nevezzük, és dom(Ai)-vel is jelöljük, n a reláciü foka.

### Hivatkozási integritási megszorítás

Egy R1 relációséma FK-val jelölt attribútumhalmaza külső (idegen) kulcsa R1-nek, amely hivatkozik az R2 relációsémára, ha eleget tesz a következő feltételeknek:

* Az FK-beli attribútumoknak és az R2 PK-val jelölt elsődleges kulcsattribútumainak páronként azonos a tartománya; ekkor azt mondjuk, hogy az FK attribútumok hivatkoznak az R2 relációsémára.
* Bármely r1 (R1) aktuális állapotának egy t1 rekordjában egy FK-beli érték vagy megjelenik egy r2 (R2) aktuális állapotának valamely t2 rekordjában PK értékeként, vagy az értéke NULL. Az előbbi esetben t1[FK]=t2[PK], ekkor azt mondjuk, hogy a t1 rekord hivatkozik a t2 rekodra.

Ha a két feltétel teljesül, egy hivatkozási integritási megszorítás áll fenn R1 –ről R2-re vonatkozóan.

### Második normálforma

Egy R relációséma második normálformában (2NF-ben) van, ha R minden másodlagos (leíró) attribútuma teljesen funkcionálisan függ R elsődleges kulcsától.

### Equjoin

Azt az általános összekapcsolási műveletet, amelynek összekapcsolási feltételében csak az egyenlőségjel (=) szerepel összehasonlító műveleti jelként, egyenlőségen alapuló összekapcsolásnak vagy más szóval equjoin műveletének nevezzük.

### Tranzakció elkülönítése (ACID)

Egy tranzakciónak látszólag más tranzakcióktól elkülönítve kell végrehajtódnia. Ez azt jelenti, hogy a tranzakció végrehajtása nem állhat kölcsönhatásban semelyik másik konkurensen végrehajtott tranzakcióval sem.

## II. rész

### 1. feladat

#### Armstrong-axióma definíciója

A reflexivitás, az augmentivitás és a tranzitivitás szabályait együtt Armstrong-Axiómának nevezzük. William Ward Armstrong 1974-ben bizonyította be, hogy a reflexivitás, az augmentivitás és a tranzivitás szabálya együtt helyes és teljes.

#### Reflexivitás bizonyítása

Tegyük fel, hogy X ⊇ Y, és hogy léteznek t1 és t2 rekordok R valamely r relációjában úgy, hogy t1[X] = t2 [X]. Ekkor t1[Y] = t2 [Y], mivel X ⊇ Y; ezért X  Y-nak teljesülnie kell r-ben.

### 2. feladat

## III. rész

### 1. feladat

#### Kapcsolattípus definíciója

Egy vagy több egyedtípus közötti jól meghatározott viszony.

#### Kapcsolattípusok osztályozása

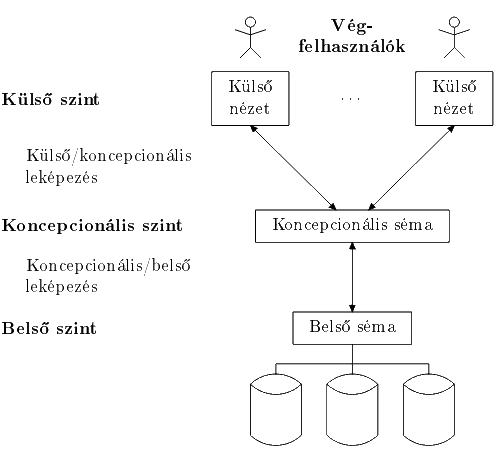
A kapcsolat foka: meghatározza, hogy hány egyedtípus vesz részt a kapcsolatban. Ez lehet bináris (másodfokú), ternáris (harmadfokú) vagy magasabb fokú.

A (bináris) kapcsolat számossága: meghatározza, hogy legfeljebb hány kapcsolat-előfordulásban vehet részt egy egyedelőfordulás. Ez lehet 1:1, 1:N, vagy M:N.

A (bináris) kapcsolat szorgossága meghatározza, hogy a kapcsolatban résztvevő egyedítpusok minden egyedének részt kell-e vennie legalább egy kapcsolat-előfordulásban. Ez lehet kötelező, félig kötelező vagy opcionális.

### 2. feladat

#### Háromséma-architectúra szintjei

A DBMS sémákat három szinten definiálja.

Külső sémák szinten a különböző felhasználói nézetek leírására. Rendszerint ugyanazt az adatmodellt használja mint a koncepcionális séma.

Koncepcionális séma koncepcionális szinten a teljes adatbázis szerkezetének és megszorításainak leírására a felhasználók közössége számára. Jellemzően koncepcionális vagy implementációs adatmodellt használ.

Belső séma belső szinten a szerkezet és az elérési utak (pl. indexek) fizikai tárolásának leírására. Jellemzően fizikai adatmodellt használ.

### 3. feladat

#### Relációalgebra

A matematikai halmazelméleten alapuló lekérdező nyelv. A lekérdezés egy kifejezés, amelyben az operátorok relációalgebrai műveletek, az operandusok pedig relációk. A lekérdezés eredménye szintén egy reláció.

#### Szelekció művelete

Általános alakja: σ(szelekciós feltétel) (R), ahol R azt a relációt jelöli, amelyből a <szelekciós feltétel>-nek eleget tevő rekordokat válogatjuk ki. A <szelekciós feltétel> pedig, egy logikai kifejezés, amely logikai operátorokkal összekapcsolt részkifejezésekből épül föl. A részkifejezések alakja lehet:

* <attribútum><hasonlító operátor><konstans>
* <attribútum><hasonlító operátor><attribútum>,

ahol az <attribútum> az R egy attribútumának neve, a <hasonlító operátor> a {=, ≠,<, >, ≤ ,≥ } operátorok egyike, a <konstans> pedig egy konstans érték az attribútum tartományból.

Egy általános szelekciós feltételben a részkifejezeséket az és, a vagy és a nem logikai operátorokkal kapcsolhatjuk össze.

Szelekció megvalósítása SQL-ben: SELECT \* FROM R WHERE szelekciós\_feltétel;

#### Szelekció tulajdonságai

A szelekció unáris művelet. Az eredményül kapott reláció foka és sémája megegyezik R fokával, illetve sémájával. A számossága pedig mindig kisebb vagy egyenlő R számosságánál, azaz bármely f feltétel esetén |σf(R)|≤ |R|.

Két egymásba ágyaztatott szelekciós művelet végrehajtási sorrendje felcserélhető. Minden többszörösen egymásba ágyazott szelekció átírható egyetlen szelekcióvá, amelynek a feltétele az eredeti feltételek konjukciója.

### 4. feladat

#### Relációs adatbázis-tervezés

Az attribútumok csoportosítás olyan módon, hogy „jó” relációsémákat alkossanak.

#### Relációs adatbázis-tervezés nem hivatalos irányelvei

##### 1. A reláció attribútumának szemantikája

Egy reláció minden egyes rekordja egy egyed-előfordulást vagy kapcsolat-előfordulást reprezentáljon. (Az egyes relációkra és azok attribútumaira külön-külön vonatkozik).

* Különböző egyedek (DOLGOZÓ-k, OSZTÁLY-ok, PROJEKT-ek) attribútumai nem keverendőek egyazon relációban.
* Más egyedekre való hivatkozás csak külső kulcsok használatával történjen.
* Az egyedekre és a kapcsolatokra vonatkozó attribútumokat a lehető legjobban el kell különíteni egymástól.

Összefoglalva: Olyan sémát kell tervezni, ami könnyen magyarázható relációról relációra. Az attribútumok szemantikájának könnyen értelmezhetőnek kell lennie.

Amikor az információt redundánsan tároljuk, az vagy tárhelyet pazarol, vagy karbantartási anomáliákat (beszúrási -, törlési-, módosítási anomália) okoz.

##### 2. Redundáns információk a rekordokban és karbantartási anomáliák

Olyan sémát tervezzünk, amelyben nem jelennek meg karbantartási anomáliák. Ha mégis előfordulnak anomáliák, akkor jegyezzük fel azokat, hogy az alkalmazások számításba vehessék őket.

* Módosítási anomália: Ha megváltoztatjuk a projekt nevét, akkor minden rajta dolgozó ügyfélnél át kell írni az új projektnévre.
* Beszúrási anomália: Nem lehet új projektet beszúrni, ha nincs hozzá dolgozó rendelve, és nem lehet új dolgozót hozzárendelni egy nem létező projekthez.
* Törlési anomália: Ha törlünk egy projektet, akkor az összes hozzátartozó dolgozót is törölni kell.

##### 3. NULL értékek a rekordban

A relációkat úgy kell megtervezni, hogy a rekordjaik a lehető legkevesebb NULL értéket tartalmazzák. Azok az attribútumok, amelyek gyakran vesznek fel NULL értéket, külön relációba tehetők (az elsődleges kulccsal). A NULL érték okai lehetnek:

* Az attribútum nem értelmezhető vagy érvénytelen.
* Az attribútumérték ismeretlen (de létezik).
* Az érték biztosan létezik, de nem elérhető.
* A relációs adatbázisok rossz tervezése bizonyos összekapcsolási műveletek esetén hibás eredményhez vezethet.
* A „veszteségmentes összekapcsolás” tulajdonsággal garantáljuk, hogy az összekapcsolási műveletek értelmes eredményt adnak.

##### 4. Álrekord

A relációkat úgy kell megtervezni, hogy kielégítsék a veszteségmentes összekapcsolás feltételét. Egy tetszőleges, relációkon végrehajtott természetes összekapcsolás nem állíthat elő álrekordokat.

### 5. feladat

#### Tranzakció definíciója

Adatbázis folyamatok egy olyan logikai egysége, amely egy vagy több adatbázis-hozzáférési műveletet (olvasás – kinyerés, írás – beszúrás, frissítés, törlés) tartalmaz.

#### Tranzakció állapotai

* Aktív állapot
* Részlegesen véglegesített állapot
* véglegesített (commit) állapot
* hibás állapot
* megszakított állapot

### 6. feladat

#### Tranziencia OO adatbázisokban

Az objektumok az OO programozási nyelvekben tranziensek (átmenetiek), a futás befejeztével törlődnek.

#### Perzisztencia OO adatbázisokban

Az objektumok az O(R)DMS-ekben perzisztensek (állandóak), eltárolódnak, később kinyerhetőek és megoszthatóak más programokkal.

### 7. feladat

#### NoSQL adatmodellek

* Kulcs-érték modell
* Rendezett kulcs-érték modell
* Oszlopcsalád modell
* Dokumentum modell
* Gráfmodell

# Feladatok – 2023.05.31

## I. rész

Definiálja a következő fogalmakat!

* Koncepcionális adatmodell
* Gyenge egyedtípus
* Atomikusság (ACID)
* Projekció
* Kulcs

## II. rész

1. Definiálja az Armstrong-axiómákat, és azok közül bizonyítsa be a tranzitivitás szabályát!
2. Képezze le az alábbi ER diagramot a LÉGITÁRSASÁG adatbázisról relációs modellé. Indokolja is meg lépéseit

## III. rész

1. Definiálja a program-adat függetlenséget!
2. Ismertesse a lehetséges integritási megszorítási sérüléseket!
3. Definiálja az uniókompatibilitás fogalmát és ismertesse a relációalgebra halmazműveleteit és azok tulajdonságait!
4. Sorolja fel az ER modell leképzésének lépéseit relációs modellé!
5. Ismertesse azokat a problémákat, amelyek tranzakciók egyidejű feldolgozásakor léphetnek fel!
6. Hogyan örököltethet felhasználó által definiált típust és mik ennek a tulajdonságai?
7. Definiálja a gyenge és az erős konzisztenciát NoSQL adatbázisokban

# Megoldások – 2023.05.31

## I. rész

### Koncepcionális adatmodell

Véges számú tulajdonságtípussal megadott véges számú egyedtípus és a közöttük fennálló véges számú kapcsolattípus összessége.

A kapcsolattípus is rendelkezhet véges számú tulajdonságtípussal.

### Gyenge egyedtípus

Azokat az egyedtípusokat, amelyek nem rendelkeznek saját kulcsattribútumokkal, gyenge egyedtípusnak nevezzük.

A gyenge egyedtípusoknak részleges kulcsuk (diszkriminátoruk) van, amely azon attribútumok halmaza, amelyek egyértelműen azonosítják azokat a gyenge egyedeket, amelyek ugyanazon tulajdonos egyed(ek)hez kapcsolódnak.

### Atomikusság (ACID)

A tranzakció a feldolgozás atomi egysége; vagy teljes egészében végrehajtódik, vagy egyáltalán nem.

### Projekció

Általános alakja: π⟨attribútumlista⟩(R), ahol az <attribútumlista> az R reláció lekérdezni kívánt attribútumainak listája.

A projekció megvalósítása SQL-ben: SELECT attribútumlista FROM R;, ahol az attribútumlista elemeit vesszővel választjuk el és a tulajdonos reláció azonosítására, ha több relációt sorolunk fel, alkalmazható a pontozott jelölés R.A

### Kulcs

Egy R relációséma K kulcsa R-nek egy olyan szuperkulcsa, amelyből bármelyik A attribútumot elhagyva, az így kapott K’ = K \ A attribútumhalmaz már nem szuperkulcsa R-nek. Egy kulcs kielégíti a következő feltételeket:

* Bármilyen relációt tekintve, a reláció két különböző rekordjának nem lehetnek azonosak a kulcsban szereplő attribútumokhoz tartozó értékek.
* Minimális szuperkulcs, azaz egy olyan szuperkulcs, amelyből nem tudunk úgy eltávolítani egyetlen attribútumot sem, hogy az egyediségre vonatkozó feltétel továbbra is fennálljon.

## II. rész

### 1. feladat

#### Armstrong-axióma definíciója

A reflexivitás, az augmentivitás és a tranzitivitás szabályait együtt Armstrong-Axiómának nevezzük. William Ward Armstrong 1974-ben bizonyította be, hogy a reflexivitás, az augmentivitás és a tranzivitás szabálya együtt helyes és teljes.

#### Tranzitivitás szabálya

Tegyük fel, hogy (1) XY és (2) Y  Z fennáll egy r relációban. Ekkor tetszőleges t1 és t2 r-beli rekordokra, melyekre igaz, hogy t1[X] = t2[X], (1) miatt kapjuk, hogy t1[Y] = t2[Y]; így (3)-ból és a (2)-es feltevésünkből azt is kapnunk kell, hogy t1[Z] = t2[Z]; ezért X  Z-nek fenn kell állnia r-ben.

### 2. feladat

## III. rész

### 1. feladat

#### Program-adat függetlenség

A programok és adatok elkülönítését program-adat függetlenségnek nevezzük. Ez lehetővé teszi az adatszerkezetek és a tárolás módjának megváltoztatását anélkül, hogy a DBMS-t elérő programot meg kellene változtatni.

##### Logikai adatfüggetlenség

Annak képessége, hogy a koncepcionális séma anélkül változzon meg, hogy a külső sémának és a hozzájuk rendelt alkalmazói programoknak meg kellene változnia.

##### Fizikai adatfüggetlenség

Annak képessége, hogy a belső séma anélkül változzon meg, hogy a koncepcionális sémának meg kellene változnia. Pl. a belső séma megváltozhat anélkül, hogy bizonyos fájl szerkezeteket átszervezünk vagy új indexeket hozunk létre az adatbázis hatékonyságának a javítása miatt.

##### Alacsonyabb, magasabb szintű változás

Amikor egy alacsonyabb szintű séma megváltozik, akkor csak ez és az eggyel magasabb szintű sémák közötti leképzésnek kell változnia.

A magasabb szintű sémák változatlanok maradnak. Ezért az alkalmazói programoknak nem szükséges módosulniuk, mivel azok a külső sémákra hivatkoznak.

### 2. feladat

#### Integritási megszorítási sérülések

##### INSERT művelet

* tartomány megszorítás, ha az új rekord egyik attribútum értéke nem a megadott tartományba esik
* kulcs megszorítás, ha az új rekord kulcs attribútum értéke már létezik a reláció egy másik rekordjánál
* hivatkozási integritás, ha a külső kulcs érték az új rekordban egy olyan elsődleges kulcs értékre hivatkozik, amelyen nem létezik a hivatkozott relációban
* egyedintegritás, ha az elsődleges kulcs érték NULL az új rekordban

##### DELETE művelet

Csak hivatkozási integritási megszorításnál okozhat sérülést, ha olyan elsődleges kulcs értékkel bíró rekordot törlünk, amelyre más relációból hivatkozás van.

##### UPDATE művelet

A tartomány megszorítás és a NULL érték megszorítás sértheti meg.

### 3. feladat

#### Uniókompatibilitás fogalma

Az R(A1, A2, … An) és S(B1, B2, … Bn) relációkat egymással uniókompatibilisnek (típuskompatibilisnek) mondjuk, ha azonos fokszámuk és dom(Ai) = dom(Bi) minden 1 ≤ i ≤ n esetén. Azaz az uniókompatibilitás azt jelenti, hogy a két relációnak ugyanannyi attribútuma van, és attribútumaik tartományai páronként megegyeznek egymással.

#### Relációalgebra halmazműveletei és tulajdonságai

##### Halmazműveletek

* unió R ∪ S : azok a rekordok, amelyek szerepelnek valamelyik relációban.
* metszet R ∩ S: azok a rekordok, amelyek mindkét relációban szerepelnek.
* különbség R – S: azok a rekordok, amelyek szerepelnek az első relációban de nem szerepelnek a másodikban

##### Tulajdonságok

Az unió, a metszet és a különbség bináris műveletek. Az eredményül kapott reláció sémája – megállapodás szerint – az első (R) reláció sémájával egyezik meg.

Az unió és a metszet műveletek kommutatívak: R ∪ S = S ∪ R és R ∩ S = S ∩ R.

Az unió és a metszet műveletek asszociatívak: R ∪ (S ∪ T) = (R ∪ S) ∪ T és R ∩ (S∩T) = (R ∩ S) ∩ T.

A különbség művelet általában nem kommutatív: R – S ≠S – R.

### 4. feladat

#### ER modell -> relációs modell

1. Erős egyedtípusok leképezése
2. Gyenge egyedtípusok leképezése
3. Bináris 1:1 számosságú kapcsolattípusok leképezése
   1. külső kulcs használata
   2. összevonás
   3. kereszthivatkozás v. kapcsoló reláció használata
4. Bináris 1:N számosságú kapcsolattípusok leképezése
5. Bináris M:N számosságú kapcsolattípusok leképezése
6. Többértékű attribútumok leképezése
7. N-edfokú kapcsolattípusok leképezése

### 5. feladat

#### Problémák amelyek tranzakciók egyidejű feldolgozásakor léphetnek fel

##### Elveszett frissítés

Akkor fordul elő, amikor két tranzakció, amely ugyanazokat az adatbázis elemeket éri el, úgy fésülődik össze, hogy egyes adatbázis elemek hibásakká válnak.

##### Időleges frissítés (dirty read)

Akkor fordul elő, amikor egy tranzakció frissít egy adatbázis elemet, ami után valamilyen oknál fogva a tranzakció hibásan fejeződik be. Ezt a frissített elemet más tranzakció is eléri mielőtt az még visszaállna az eredeti értékre.

##### Helytelen összegzés

Amikor egy tranzakció rekordok egy összegző függvényét számolja, amíg egy másik tranzakció ezen rekordok közül néhányat frissít. Ekkor az összegző függvény olyan értékekkel számolhat, amelyek még a frissítés előtt vannak, míg mások már a frissítés után.

### 6. feladat

#### Felhasználó által definiált típus célja

Összetett szerkezetű (a relációs modell rekordjainál bonyolultabb) objektumok létrehozása. Egy típus deklarációjának elválasztása a tábla (reláció) lérehozásától. Rekord típusú konstruktor a ROW kulcsszóval rekord típusú attribútumok létrehozására.

4 féle kollekció típus: ARRAY, MULTISET, LIST és SET

#### Öröklődés

SQL-ben UNDER kulcsszó alatt működik

A NOT FINAL kulcsszót kell használni ha egy UDT-nek további altípusát szeretnénk deklarálni.

Minden attribútum öröklődik.

A szupertípusok sorrendje az UNDER kulcsszó után határozza meg az öröklődési sorrendet.

Egy altípus példánya minden olyan kontextusban használható ahol szupertípusának példánya használható.

Egy altípus minden a szupertípuson definiált függvényt újradefiniálhat feltéve hogy a szignatúra nem változhat.

Egy függvény hívásakor a legjobban illeszkedő implementáció kerül alkalmazásra az összes argumentum típusát figyelembe véve.

Dinamikus kötéskor a paraméterek futáskori típusait veszi figyelembe

### 7. feladat

#### Gyenge és erős konzisztencia NoSQL-ben

##### Gyenge konzisztencia

A rendszer nem garantálja, hogy a későbbi hozzáférések a módosított értéket adják eredményül. Több feltételnek is teljesülnie kell, mielőtt az érték visszaadásra kerül. A módosítás megtörténte és azon pillanat közötti időszakot, amelyre már garantált, hogy minden megfigyelő mindig a módosított értéket látja, inkonzisztencia ablaknak (inconsistency window) nevezzük.

##### Erős konzisztencia

Miután a módosítás végrehajtódott, minden (akár A, akár B, akár C által végzett) hozzáférés a módosított értéket adja eredményül.

# Feladat – 2023.06.12

## I. rész

Definiálja következő fogalmakat!

* Adatbázis
* Kapcsolattípus
* Funkcionális függés
* Konzisztencia (ACID)
* Uniókompatibilitás

## II. rész

1. Definiálja az Armstrong-axiómákat, és azok közül bizonyítsa be az augmentivitás szabályát!
2. Képezze le az alábbi ER diagramot a MÚZEUM adatbázisról relációs modellé! Indokolja is meg a lépéseit!

## III.rész

1. Osztályozza a tulajdonságtípusokat (attribútumokat)!
2. Ismertesse az adatbázis séma és állapot fogalmakat!
3. Ismertesse a relációalgebra projekció műveletét és tulajdonságait!
4. Sorolja fel az EER séma leképzésének műveletét és tulajdonságait!
5. Sorolja fel a tranzakciós műveleteket!
6. Ismertesse az adattárházak és a hagyományos adatbázisok közötti különbségeket!

# Megoldások – 2023.06.12

## I. rész

### Adatbázis

Az adatmodell, valamint az egyed-előfordulások, tulajdonság-előfordulások és kapcsolat-előfordulások együttese.

### Kapcsolattípus

Két vagy több egyedtípus közötti jól meghatározott viszony.

### Funkcionális függés

Az R két attribútumhalmaza, X és Y között, XY-nal jelölt funkcionális függés előír egy megszorítást azokra a lehetséges rekordokra, amelyek egy R fölötti r relációt alkothatnak. A megszorítás az, hogy bármely két, r-beli t1 és t2 rekord esetén, amelyekre t1[X]=t2[X] teljesül, teljesülnie kell t1[Y]=t2[Y]-nak is. – Azaz egy R relációsémában X akkor és csak akkor határozza meg funkcionálisan Y-t, ha valahányszor r(R) két rekordja megegyezik az X értékeken, szükségszerűen megegyezik az Y értékeken is.

### Konzisztencia (ACID)

Konzisztenciamegőrzés (consistency preservation): Egy tranzakció konzisztenciamegőrző, ha teljes és önálló végrehajtása az adatbázist konzisztens állapotból konzisztens állapotba viszi át.

### Uniókompatibilitás

Az R(A1, A2, … An) és S(B1, B2, … Bn) relációkat egymással uniókompatibilisnek (típuskompatibilisnek) mondjuk, ha azonos fokszámuk és dom(Ai) = dom(Bi) minden 1 ≤ i ≤ n esetén. -- Azaz az uniókompatibilitás azt jelenti, hogy a két relációnak ugyanannyi attribútuma van, és attribútumaik tartományai páronként megegyeznek egymással.

## II. rész

### 1. feladat

#### Armstrong-axióma definíciója

A reflexivitás, az augmentivitás és a tranzitivitás szabályait együtt Armstrong-Axiómának nevezzük. William Ward Armstrong 1974-ben bizonyította be, hogy a reflexivitás, az augmentivitás és a tranzivitás szabálya együtt helyes és teljes.

#### Augmentivitás szabályának bizonyítása

Tegyük fel, hogy XY fennáll R egy r relációjában, de XZ  YZ nem áll fenn. Ekkor léteznie kell t1 és t2 rekordoknak úgy, hogy

1. t1[X] = t2[X],
2. t1[Y] = t2[Y],
3. t1[XZ] = t2[XZ] és
4. t1[YZ] ≠t2[YZ].

Ez nem lehetséges, mert (3)-nól kapjuk, hogy (5) t1[Z] = t2[Z], míg (2)-ből és (5)-ből kapjuk, hogy (6)t1[YZ] = t2[YZ], ami ellentmond (4)-nek.

### 2. feladat

## III. rész

### 1. feladat

#### Tulajdonságtípus

Az azonos szerepű tulajdonságok absztrakciója. A tulajdoságtípus egy másik elnevezése az attribútum.

#### Tulajdonságtípus attribútumainak osztályozása

A tulajdonság-előfordulás szerkezete (összetettsége) szerint lehet egyszerű/atomi (pl. születési hely) vagy összetett (pl. lakcím).

A tulajdonság-előfordulás hány értéket vehet fel egyszerre ez lehet egyértékű (pl. születési dátum), halmazértékű vagy többértékű (pl. felvett kurzusok).

A tulajdonság-előfordulás minden esetben megjelenik-e a háttértárolón (a fizikai adatbázisban) ez lehet tárolt vagy származtatott.

### 2. feladat

#### Adatbázis séma fogalma

Az adatbázis séma az adatbázis leírása. Az adatbázis szerkezetének, az adattípusoknak és a megszorításoknak a leírását tartalmazza.

#### Adatbázis állapot fogalma

Az adatbázis állapot az adatbázis egy időpillantbeli tartalmát jelenti. Egy időpillanatban az adatbázisban tárolt aktuális adatok összessége. Nevezik az adatbázis egy előfordulásának.

### 3. feladat

#### Relációalgebra

A matematikai halmazelméleten alapuló lekérdező nyelv. A lekérdezés egy kifejezés, amelyben az operátorok relációalgebrai műveletek, az operandusok pedig relációk. A lekérdezés eredménye szintén egy reláció.

#### Reláció algebra projekció művelete

Általános alakja: π⟨attribútumlista⟩(R), ahol az <attribútumlista> az R reláció lekérdezni kívánt attribútumainak listája.

A projekció megvalósítása SQL-ben: SELECT attribútumlista FROM R;, ahol az attribútumlista elemeit vesszővel választjuk el és a tulajdonos reláció azonosítására, ha több relációt sorolunk fel, alkalmazható a pontozott jelölés R.A

#### Projekció tulajdonságai

A projekció unáris művelet. Az eredményül kapott reláció fokát és sémáját az attribútumlistában szereplő attribútumok határozzák meg. Az eredmény sémájában az attribútumok sorrendje megegyezik a listában megadott attribútumok sorrendjével. A fokszám pedig a listában megadott attribútumok darabszáma.

Ha az attribútum lista nem tartalmaz kulcs attribútumot, akkor az eredményül kapott reláció számossága kisebb lehet R számosságánál, ugyanis az eredményekben nem jelenhetnek meg duplikált rekordok. Ha az attribútumlista R szuperkulcs, akkor az eredmény számossága megegyezik R számosságával.

Két ágymásba ágyazott projekciós művelet eredménye megegyezik a külső projekció eredményével: πlista1(πlista2(R)) = πlista1(R), ha lista2 ⫆ lista1, egyébként a baloldal nem értelmezhető.

### 4. feladat

#### EER séma leképzése relációs sémává

1. Erős egyedtípusok leképezése
2. Gyenge egyedtípusok leképezése
3. Bináris 1 : 1 számosságú kapcsolattípusok leképezése
   1. külső kulcs használata
   2. összevonás
   3. kereszthivatkozás v. kapcsoló reláció használata
4. Bináris 1 : N számosságú kapcsolattípusok leképezése
5. Bináris M : N számosságú kapcsolattípusok leképezése
6. Többértékű attribútumok leképezése
7. N-edfokú kapcsolattípusok leképezése
8. Specializációk és generalizációk leképezése
9. Unió típusok (kategóriák) leképezése

### 5. feladat

#### Tranzakciós műveletek

* begin\_transaction
* read vagy write
* end\_transaction
* commit\_transaction
* rollback vagy abort

### 6. feladat

#### Felhasználó által definiált típus (UDT)

##### Célja

Összetett szerkezetű (a relációs modell rekordjainál bonyolultabb) objektumok létrehozása. Egy típus deklarációjának elválasztása a tábla (reláció) lérehozásától. Rekord típusú konstruktor a ROW kulcsszóval rekord típusú attribútumok létrehozására.

4 féle kollekció típus: ARRAY, MULTISET, LIST és SET

##### Létrehozása SQL-be

CREATE TYPE típus\_neve AS (komponensek deklarációja)

##### További tulajdonságok

Objektumok egyértelmű azonosítása, rendszer által generált OID-vel referencia típus útján: REF IS SYSTEM GENERATED. Emelett használható a relációs modell hagyományos kulcsa is.

A példányosítható (INSTANTIABLE kulcsszó) UDT-khez táblákat (relációkat) is létrehozhatunk.

Az UDT-khez műveleteket (metódusokat) is definiálhatunk.

Attribútumok és műveletek három fajtája:

* PUBLIC - látható az UDT interfészen
* PRIVATE - nem látható az UDT interfészen
* PROTECTED - csak az altípusok számára látható

### 7. feladat

#### Adattárházak és hagyományos adatbázisok közötti különbség

Az adattárházakat főként a gyors adatelérésre optimalizálják. A hagyományos adatbázisok tranzakciósak egyaránt optimalizáltak az adatelérési mechanizmusok és a konzisztencia biztosítása tekintetében.

Az adattárházak nagyobb hangsúlyt helyeznek a historikus adatokra mivel fő céljuk idősorok és trend elemzések támogatása.

A tranzakciós adatbázisokkal szemben az adattárházak nem változnak abban az értelemben, hogy ha egy adat egyszer oda bekerült, akkor az ott is marad változatlan formában az „idők végezetéig”

A tranzakciós adatbázisokban a tranzakció az a mechanizmus, amely megváltoztatja az adatbázist. Ezzel szemben az adattárházakban az információ durván szemcsézett és a frissítési politika alaposan megválasztott általában inkrementális jellegű.

# Feladatok – 2023.06.20

## I. rész

Definiálja a következő fogalmakat!

* Relációs adatbázisséma
* Szuperkulcs
* Harmadik normálforma
* Adattárház
* Tranzakció tartóssága (ACID)

## II. rész

1. Sorolja fel és definiálja az Armstrong-axiómákat, bizonyítsa be a tranzivitás szabályát!
2. A FITNESS adatbázis az alábbi relációssémákból áll:
   1. EDZŐTEREM(terem\_azon, név, cím, alapterület)
   2. ÜGYFÉL(szemszám, név, szül\_dátum, lakcím, testsúly)
   3. TRÉNER(edzőterem\_azon, név, szül\_dátum, lakcím)
   4. BÉRLET(bérlet\_szám, ügyfél\_azon, edzőterem\_azon, edzés\_típus)
   5. EDZÉS(ügyfél\_azon, edzőterem\_azon, dátum)

Minek felelnek meg az egyes relációsémák az ER modelben (erős, gyenge egyedtípus illetve kapcsolattípus)? Adja meg értelemszerűen az egyes relációsémákhoz tartozó integritási megszorításokat (kulcs, külső kulcs, NULL, egyed stb.)! Ábrázolja az adatbázist ER sémával (ügyeljünk arra, hogy egy ügyfél mindig ugyanabba az edzőterembe jár)! Milyen normálformában vannak a relációsémák és miért?

## III. rész

1. Ismertesse a logikai és fizikai adatfüggetlenséget!
2. Ismertesse a relációalgebra unáris műveleteit (definíció és tulajdonságok)!
3. Ismertesse a relációs modell megszorításait, azon belül sorolja fel a sémaalapú megszorításokat!
4. Sorolja fel az ER séma leképzésének lépéseit relációs sémává
5. Sorolja fel a karbantartási anomáliákat és mondjon rájuk példákat!
6. Definiálja egy tranzakció véglegesítési (commit) pontját!
7. Mi a különbség a többszörös és a szelektív öröklődés között?

# Megoldások – 2023.06.20

## I. rész

### Relációs adatbázisséma

Egy S relációs adatbázisséma az S = {R1, R2, …, Rm} relációséma-halmaz, valamint integritási megszorítások – IC-vel jelölt-halmazának az együttese.

### Szuperkulcs

Az R relációsémának létezik egy olyan attribútumhalmaza, amelyen olyan tulajdonságú, hogy tekintve R bármelyik r relációját, az adott relációban nincs két olyan rekord, amelynek az értékei azonosak lennének ezen attribútumokra vonatkozóan. Az attribútumoknak egy ilyen részhalmazát SK-val jelölve, bármely két különböző t1 és t2 rekordot kiválasztva R egy r relációjából:

t1[SK] ≠t2[SK].

Minden ilyen SK attribútumhalmaz az R relációséma szuperkulcsa. Minden relációnak van legalább egy szuperkulcsa – az összes attribútumának a halmaza.

### Harmadik normálforma

A harmadik normálforma a tranzitív függés szabályán alapul. Egy R relációséma harmadik normálformában (3 NF-ben) van, ha 2NF-ben van, és nincs R-nek olyan másodlagos (leíró) attribútuma, amely tranzitívan függne az elsődleges kulcstól.

### Adattárház

Az adattárház adatok téma-orientált, integrált, nemváltozó, időbélyeggel rendelkező összessége a menedzsment döntéseinek támogatására.

### Tranzakció tartóssága (ACID)

Tartósság vagy állandóság (durability vagy permanency): Egy véglegesített tranzakció által az adatbázison véghezvitt módosításoknak meg kell őrződniük az adatbázisban. Ezeknek a módosításoknak semmilyen hiba miatt nem szabad elveszniük.

## II. rész

### 1. feladat

#### Armstrong-axióma definíciója

A reflexivitás, az augmentivitás és a tranzitivitás szabályait együtt Armstrong-Axiómának nevezzük. William Ward Armstrong 1974-ben bizonyította be, hogy a reflexivitás, az augmentivitás és a tranzivitás szabálya együtt helyes és teljes.

#### Tranzitivitás szabálya

Tegyük fel, hogy (1) XY és (2) Y  Z fennáll egy r relációban. Ekkor tetszőleges t1 és t2 r-beli rekordokra, melyekre igaz, hogy t1[X] = t2[X], (1) miatt kapjuk, hogy t1[Y] = t2[Y]; így (3)-ból és a (2)-es feltevésünkből azt is kapnunk kell, hogy t1[Z] = t2[Z]; ezért X  Z-nek fenn kell állnia r-ben.

### 2. feladat

## III. rész

### 1. feladat

#### Program-adat függetlenség

A programok és adatok elkülönítését program-adat függetlenségnek nevezzük. Ez lehetővé teszi az adatszerkezetek és a tárolás módjának megváltoztatását anélkül, hogy a DBMS-t elérő programot meg kellene változtatni.

##### Logikai adatfüggetlenség

Annak képessége, hogy a koncepcionális séma anélkül változzon meg, hogy a külső sémának és a hozzájuk rendelt alkalmazói programoknak meg kellene változnia.

##### Fizikai adatfüggetlenség

Annak képessége, hogy a belső séma anélkül változzon meg, hogy a koncepcionális sémának meg kellene változnia. Pl. a belső séma megváltozhat anélkül, hogy bizonyos fájl szerkezeteket átszervezünk vagy új indexeket hozunk létre az adatbázis hatékonyságának a javítása miatt.

### 2. feladat

#### Unáris műveletek

##### Szelekció

Általános alakja: σ(szelekciós feltétel) (R), ahol R azt a relációt jelöli, amelyből a <szelekciós feltétel>-nek eleget tevő rekordokat válogatjuk ki. A <szelekciós feltétel> pedig, egy logikai kifejezés, amely logikai operátorokkal összekapcsolt részkifejezésekből épül föl. A részkifejezések alakja lehet:

* <attribútum><hasonlító operátor><konstans>
* <attribútum><hasonlító operátor><attribútum>,

ahol az <attribútum> az R egy attribútumának neve, a <hasonlító operátor> a {=, ≠,<, >, ≤ ,≥ } operátorok egyike, a <konstans> pedig egy konstans érték az attribútum tartományból.

Egy általános szelekciós feltételben a részkifejezeséket az és, a vagy és a nem logikai operátorokkal kapcsolhatjuk össze.

Szelekció megvalósítása SQL-ben: SELECT \* FROM R WHERE szelekciós\_feltétel;

###### Szelekció tulajdonságai

A szelekció unáris művelet. Az eredményül kapott reláció foka és sémája megegyezik R fokával, illetve sémájával. A számossága pedig mindig kisebb vagy egyenlő R számosságánál, azaz bármely f feltétel esetén |σf(R)|≤ |R|.

Két egymásba ágyaztatott szelekciós művelet végrehajtási sorrendje felcserélhető. Minden többszörösen egymásba ágyazott szelekció átírható egyetlen szelekcióvá, amelynek a feltétele az eredeti feltételek konjukciója: σfelt1 (σfelt2 (…(σfeltn (R))…)) = σfelt1 AND felt2 AND... AND feltn (R)

##### Projekció

Általános alakja: π⟨attribútumlista⟩(R), ahol az <attribútumlista> az R reláció lekérdezni kívánt attribútumainak listája.

A projekció megvalósítása SQL-ben: SELECT attribútumlista FROM R;, ahol az attribútumlista elemeit vesszővel választjuk el és a tulajdonos reláció azonosítására, ha több relációt sorolunk fel, alkalmazható a pontozott jelölés R.A

A projekció unáris művelet. Az eredményül kapott reláció fokát és sémáját az attribútumlistában szereplő attribútumok határozzák meg. Az eredmény sémájában az attribútumok sorrendje megegyezik a listában megadott attribútumok sorrendjével. A fokszám pedig a listában megadott attribútumok darabszáma.

Ha az attribútum lista nem tartalmaz kulcs attribútumot, akkor az eredményül kapott reláció számossága kisebb lehet R számosságánál, ugyanis az eredményekben nem jelenhetnek meg duplikált rekordok. Ha az attribútumlista R szuperkulcs, akkor az eredmény számossága megegyezik R számosságával.

Két ágymásba ágyazott projekciós művelet eredménye megegyezik a külső projekció eredményével: πlista1(πlista2(R)) = πlista1(R), ha lista2 ⫆ lista1, egyébként a baloldal nem értelmezhető.

##### Átnevezés

Általános alakja: ρS(B1,B2,...,Bn)(R) vagy ρS (R) vagy ρ(B1,B2,...,Bn)(R), ahol az S a reláció jelölésére használt új szimbólum és B1,B2,...,Bn az új attribútumnevek.

Az átnevezés unáris művelet. Az eredményül kapott reláció foka és számossága megegyezik R fokával, illetve számosságával. Az eredményül kapott reláció sémája a B1,B2,...,Bn attribútumokkal meghatározott séma lesz, ha megadtuk őket, megegyezik az R sémájával, ha a B1,B2,...,Bn attribútumokat nem soroltuk fel.

### 3. feladat

#### Relációs modell megszorításai

Az adatmodellben benne rejlő megszorítások: modellalapú vagy implicit megszorítások (pl. az adatok egy matematikai relációba szerveződnek holott alkothatnának más struktúrát is.)

Az adatmodell sémáiban közvetlenül kifejezett megszorítások: sémalapú vagy explicit megszorítások.

OIyan megszorítások, amelyeket nem lehet közvetlenül az adatmodell sémáiban kifejezni, és ezért az alkalmazói programokkal kell kifejezni, és érvényre juttatni őket: alkalmazásalapú vagy szemantikus megszorítások vagy üzleti szabályok.

#### Séma alapú megszorítások

##### Tartománymegszorítás

A tartománymegszorítás kimondja, hogy minden rekordban minden egyes A attribútumhoz tartozó érték a dom(A) tartományból származik vagy NULL érték, és ezen dom(A) tartományok minden elemének atomi értéknek kell lennie.

##### Kulcsmegszorítás

A kulcsmegszorítás szerint a relációsémának mindig rendelkeznie kell elsődleges kulccsal. A PRIMARY KEY megszorítás a NOT NULL és UNIQUE megszorítások kombinációja. A UNIQUE megszorítás tiltja ugyanazon atomi érték többszöri használatát egy attribútumnál, viszont a NULL értéket nem, azaz az többször is előfordulhat.

##### NULL értékre vonatkozó megszorítás

Egyetlen elsődlegeskulcs-érték sem lehet NULL érték.

##### Egyedintegritási megszorítás

Az egyedintegritási megszorítás kimondja, hogy egyetlen elsődlegeskulcs-érték sem lehet NULL érték. Ha az elsődleges kulcs összetett, akkor annak egyik komponense sem lehet NULL érték.

##### Hivatkozási integritási megszorítás

A hivatkozási integritási megszorítást két reláció között értelmezzük, és a két relációban lévő rekordok közötti konzisztencia megteremtése érdekében használjuk.

### 4. feladat

#### ER séma -> relációs séma

1. Erős egyedtípusok leképezése
2. Gyenge egyedtípusok leképezése
3. Bináris 1:1 számosságú kapcsolattípusok leképezése
   1. külső kulcs használata
   2. összevonás
   3. kereszthivatkozás v. kapcsoló reláció használata
4. Bináris 1:N számosságú kapcsolattípusok leképezése
5. Bináris M:N számosságú kapcsolattípusok leképezése
6. Többértékű attribútumok leképezése
7. N-edfokú kapcsolattípusok leképezése

### 5. feladat

#### Karbantartási anomáliák

##### Módosítási anomália

Ha megváltoztatjuk a projekt nevét, akkor minden rajta dolgozó ügyfélnél át kell írni az új projektnévre.

##### Beszúrási anomália

Nem lehet új projektet beszúrni, ha nincs hozzá dolgozó rendelve, és nem lehet új dolgozót hozzárendelni egy nem létező projekthez.

##### Törlési anomália

Ha törlünk egy projektet, akkor az összes hozzátartozó dolgozót is törölni kell.

### 6. feladat

#### Tranzakció véglegesítési pontja

Egy tranzakció akkor éri el a véglegesítési (commit) pontját, ha az összes adatbázis-hozzáférési művelete sikeresen végrehajtódott és ezen műveletek hatására kiírásra került a log-fájlba.

A véglegesítési pontja után a tranzakciót véglegesítettnek nevezzük és feltételezzük, hogy összes hatása állandó bejegyzésre került az adatbázisban. A tranzakció ezután egy [commit, T] bejegyzést tsz a log-ba.

### 7. feladat

#### Többszörös öröklődés

Akkor beszélünk róla ha egy altípus kettő vagy több típus altípusa és így értelemszerűen örökli mindkettő vagy az összes függvényét (attribútumait és metódusait).

#### Szelektív öröklődés

Amikor egy altípus csak egy típus bizonyos függvényeit örökli, amelyeket nem, azokat az EXPECT klózzal jelezzük.

# **Tartalomjegyzék**

[Feladatok – 2023.06.05 1](#_Toc167608984)

[I. rész 1](#_Toc167608985)

[II. rész 1](#_Toc167608986)

[III. rész 1](#_Toc167608987)

[Megoldások – 2023.06.05 2](#_Toc167608988)

[I. rész 2](#_Toc167608989)

[Relációséma 2](#_Toc167608990)

[Hivatkozási integritási megszorítás 2](#_Toc167608991)

[Második normálforma 2](#_Toc167608992)

[Equjoin 2](#_Toc167608993)

[Tranzakció elkülönítése (ACID) 2](#_Toc167608994)

[II. rész 2](#_Toc167608995)

[1. feladat 2](#_Toc167608996)

[2. feladat 2](#_Toc167608997)

[III. rész 2](#_Toc167608998)

[1. feladat 2](#_Toc167608999)

[2. feladat 3](#_Toc167609000)

[3. feladat 3](#_Toc167609001)

[4. feladat 4](#_Toc167609002)

[5. feladat 4](#_Toc167609003)

[6. feladat 5](#_Toc167609004)

[7. feladat 5](#_Toc167609005)

[Feladatok – 2023.05.31 6](#_Toc167609006)

[I. rész 6](#_Toc167609007)

[II. rész 6](#_Toc167609008)

[III. rész 6](#_Toc167609009)

[Megoldások – 2023.05.31 7](#_Toc167609010)

[I. rész 7](#_Toc167609011)

[Koncepcionális adatmodell 7](#_Toc167609012)

[Gyenge egyedtípus 7](#_Toc167609013)

[Atomikusság (ACID) 7](#_Toc167609014)

[Projekció 7](#_Toc167609015)

[Kulcs 7](#_Toc167609016)

[II. rész 7](#_Toc167609017)

[1. feladat 7](#_Toc167609018)

[2. feladat 7](#_Toc167609019)

[III. rész 7](#_Toc167609020)

[1. feladat 7](#_Toc167609021)

[2. feladat 8](#_Toc167609022)

[3. feladat 8](#_Toc167609023)

[4. feladat 9](#_Toc167609024)

[5. feladat 9](#_Toc167609025)

[6. feladat 9](#_Toc167609026)

[7. feladat 10](#_Toc167609027)

[Feladat – 2023.06.12 11](#_Toc167609028)

[I. rész 11](#_Toc167609029)

[II. rész 11](#_Toc167609030)

[III.rész 11](#_Toc167609031)

[Megoldások – 2023.06.12 12](#_Toc167609032)

[I. rész 12](#_Toc167609033)

[Adatbázis 12](#_Toc167609034)

[Kapcsolattípus 12](#_Toc167609035)

[Funkcionális függés 12](#_Toc167609036)

[Konzisztencia (ACID) 12](#_Toc167609037)

[Uniókompatibilitás 12](#_Toc167609038)

[II. rész 12](#_Toc167609039)

[1. feladat 12](#_Toc167609040)

[2. feladat 12](#_Toc167609041)

[III. rész 12](#_Toc167609042)

[1. feladat 12](#_Toc167609043)

[2. feladat 13](#_Toc167609044)

[3. feladat 13](#_Toc167609045)

[4. feladat 13](#_Toc167609046)

[5. feladat 14](#_Toc167609047)

[6. feladat 14](#_Toc167609048)

[7. feladat 14](#_Toc167609049)

[Feladatok – 2023.06.20 15](#_Toc167609050)

[I. rész 15](#_Toc167609051)

[II. rész 15](#_Toc167609052)

[III. rész 15](#_Toc167609053)

[Megoldások – 2023.06.20 16](#_Toc167609054)

[I. rész 16](#_Toc167609055)

[Relációs adatbázisséma 16](#_Toc167609056)

[Szuperkulcs 16](#_Toc167609057)

[Harmadik normálforma 16](#_Toc167609058)

[Adattárház 16](#_Toc167609059)

[Tranzakció tartóssága (ACID) 16](#_Toc167609060)

[II. rész 16](#_Toc167609061)

[1. feladat 16](#_Toc167609062)

[2. feladat 16](#_Toc167609063)

[III. rész 16](#_Toc167609064)

[1. feladat 16](#_Toc167609065)

[2. feladat 17](#_Toc167609066)

[3. feladat 18](#_Toc167609067)

[4. feladat 18](#_Toc167609068)

[5. feladat 19](#_Toc167609069)

[6. feladat 19](#_Toc167609070)

[7. feladat 19](#_Toc167609071)

[Tartalomjegyzék 20](#_Toc167609072)