

Adatbázisok I.

Kiterjesztett műveletek

- δ multihalmaz \rightarrow halmaz
- Összesítő műveletek:
 - SUM
 - AVG
 - MIN, MAX
 - COUNT

Csoportosítás

- a reláció sorainak „csoportokba” történő beosztása a reláció egy, vagy több attribútumának értékétől függően.
- γ (gamma) csoportosítási művelet kombinálja a csoportosítást és az összesítést.
- A γ alsó indexeként szereplő L elemeknek egy listája, ahol egy elem az alábbiak közül bármelyik lehet:
 - R reláció egy attribútuma, amelyre csoportosítást végzünk. Ezt nevezzük csoportosítási attribútumnak.
 - A reláció valamely attribútumára vonatkozó összesítési művelet. A kiindulásul szolgáló attribútumot összesítési attribútumnak nevezzük.
- Működése:
- Osszuk R sorait csoportokba.
- Egy csoport azokat a sorokat tartalmazza, amelyeknek az L listán szereplő csoportosítási attribútumokhoz tartozó értékei megegyeznek.
- Ha nincs csoportosítási attribútum, akkor az egész R reláció egy csoportot képez.
- Minden csoporthoz hozzunk létre olyan sort, amelyik tartalmazza:
 - A szóban forgó csoport csoportosítási attribútumait.
 - Az L lista összesítési attribútumaira vonatkozó összesítéseket.

Rendezés

- τ – rendezési művelet, egy relációt a sorainak egy vagy több attribútumtól függő rendezett listájává alakít.
- $\tau_L(F)$ ($L = A_1, A_2, \dots, A_n$) – F -t először A_1 , majd A_2 , majd ... A_n szerint rendezi.

Külső összekapcsolások

- FULL - Az SQL-ben a szabványos külső összekapcsolást, amely a mindkét kapcsolandó táblából származó nem kapcsolható sorokat is tartalmazza, teljes külső összekapcsolásnak nevezi.
- LEFT - Csak bal oldaliból.
- RIGHT
- Azaz a "lógó" sorokat ahelyett hogy elvetné, feltölti NULL értékkel.

SQL WITH RECURSIVE utasítás

Tegyük fel, hogy a Jaratok(legitarsasag, honnan, hova, koltseg) táblában repülőjáratok adatait tároljuk, hogy melyik városból (honnan) melyik városba (hova) van közvetlen járat. Fejezzük ki SQL WITH RECURSIVE utasítással az Eljut(honnan, hova, koltseg) átmeneti táblát, és adjuk meg hova tudunk eljutni 'Budapest'-ről legfeljebb 5000 költségből.

```
WITH RECURSIVE Eljut AS
  (SELECT honnan, hova FROM Jaratok
   UNION
   SELECT Eljut.honnan, Jaratok.hova
   FROM Eljut, Jaratok
   WHERE Eljut.hova = Jaratok.honnan)
SELECT hova
FROM Eljut
WHERE honnan = 'Bp' AND koltseg <= 5000;
```

Tranzakciók ACID tulajdonságai

- **Atomosság (atomicity):** A tranzakció a feldolgozás atomi egysége; vagy teljes

egészében végrehajtott, vagy egyáltalán nem.

- **Konzisztenciamegőrzés (consistency):** a tranzakció futása után konzisztens legyen az adatbázis, megszorításokkal, triggerekkel biztosítjuk.
- Elkülönítés (isolation): párhuzamos végrehajtás eredménye egymás utáni végrehajtással egyezzen meg.
- Tartósság vagy állandóság (durability): a befejezett tranzakció eredménye rendszerhiba esetén sem veszhet el.

Egyedhalmaz sémája és előfordulása?

- $E(A_1, \dots, A_n)$ egyedhalmaz
- séma:
 - E az egyedhalmaz neve,
 - A_1, \dots, A_n tulajdonságok,
 - $DOM(A_i)$ – lehetséges értékek halmaza. például: tanár(név, tanszék).
- előfordulása:
 - A konkrét egyedekből áll
 - $E = \{e_1, \dots, e_m\}$ egyedek (entitások) halmaza, ahol
 - $e_i(k) \in DOM(A_k)$,
 - semelyik két egyed sem egyezik meg minden attribútumban (léteznek és megkülönböztethetők)

Erős és gyenge egyedhalmaz. E/K diagramban.

- Gyenge: dupla téglalap
- Erős: téglalap
- Gyenge egyedhalmazok: Az egyedhalmazok kulcsában szereplő attribútumok közül néhány, vagy esetleg az összes, más egyedhalmaznak az attribútumai. Emiatt nem tudjuk egyértelműen meghatározni, csak a kapcsolatokkal.
- Erős: Amelyet egyértelműen tudunk azonosítani. Van azonosító jellegű tulajdonsága.

Bináris kapcsolatok

- $K(E1, E2)$
- sok-egy: K előfordulásaiban minden E1 beli egyedhez legfeljebb 1 E2 beli egyed tartozhat.
- sok-sok: minden E1 beli egyedhez több E2 beli egyed tartozhat, és fordítva, minden E2 beli egyedhez több E1 beli egyed tartozhat.
- sok-egy: minden E1 beli egyedhez legfeljebb egy E2 beli egyed tartozhat, és fordítva, minden E2 beli egyedhez legfeljebb egy E1 beli egyed tartozhat.



Alosztályok és az osztályhierarchia

- $K(E1, E2)$
- is-a: az összes E1 beli egyed szerepel (speciális egy-egy)
- jelölése: háromszögben "isa" a kettő között

Az E/K diagram relációkká való átalakításának általános elvei.

- Egyedhalmazok átírásánál vizsgáljuk meg, ha összetett típusról van szó.
- Többértékű attribútumoknál...
 - egybe vesszük az egészet
 - több sort veszünk fel egy táblába
 - külön táblát veszünk fel

E/K modell	Relációs adatmodell
egyedhalmaz séma	relációséma
tulajdonságok	attribútumok
szuperkulcs	szuperkulcs
egyedhalmaz előfordulása	reláció
egyed	sor

DML

- INSERT INTO <reláció> VALUES (<konkrét értékek listája>);
- vagy
- INSERT INTO <reláció> (<alkérdés>);
- DELETE FROM <reláció> WHERE <feltétel>;
- UPDATE <reláció> SET <attribútum értékadások listája> WHERE <sorokra vonatkozó feltétel>;

Egyszerű, összetett kulcs

- Az egyszerű kulcs egyetlen attribútumból áll.
- Az összetett kulcsot kettő vagy több oszlop kombinációja alkotja.

Hivatkozási épség megszorítás

- REFERENCES Tábla(mező) Tábla készítésekor
- A tábla egy oszlopában ha szerepel egy érték, akkor ennek B táblában is szerepelnie kell.

Összetett kulcs példa

Adja meg a következő megszorítást SQL-ben a Filmek(cím, év, hossz, stúdió) táblában a cím, év attribútumhalmaz kulcs!

```
CREATE TABLE Filmek (  
    cím VARCHAR(50)  
    , év INT  
    , hossz INT  
    , stúdió VARCHAR2  
    PRIMARY KEY (cím, év)  
);
```

Külső kulcs példa

Adja meg a hivatkozási épség megszorítást a Filmek(cím, év, hossz, stúdió) és Szerepel(filmcím, év, színésznév) tábla segítségével, ahol a Szerepel tábla (filmcím, év) attribútumpárja a Filmek tábla (cím, év) kulcsára mutat!

```
CREATE TABLE Szerepel (  
    filmcím VARCHAR(50)  
    , év INT  
    FOREIGN KEY (filmcím, év)  
    REFERENCES Filmek(cím, év)  
);
```

Attribútum (értékek) megszorítása

- NOT NULL
- CHECK(<feltétel>) hasonlóan WHERE feltételhez: alkérdést tartalmazhat, de külső tábla attribútumai csak alkérdésben szerepelhetnek.

Sorok megszorítása

- CHECK(<feltétel>)

Nézettáblák

- Előnyei:
- Korlátozható az adatok elérése
- Az adatfüggetlenség biztosítható

- Bonyolult lekérdezések egyszerűbben kifejezhetők
- Ugyanazon adatokat különböző nézőpontból tekinthetjük meg
- Két fajtája van: virtuális (nem lesz eltárolva), illetve materializált

Nézettábla létrehozása és használata

- létrehozása:

```
CREATE [OR REPLACE][FORCE|NOFORCE] [MATERIALIZED] VIEW <név> AS <lekérdezés>
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

- törlés:

```
DROP VIEW <név>;
```

- lekérdezésük ugyanúgy történik mint az alaptábláké.
- bizonyos esetben módosíthatóak az alaptáblák is a nézettáblákon keresztül.

PL/SQL függvények

```
CREATE [OR REPLACE] FUNCTION <függvéynév>(<paraméterek>)]
    RETURN <visszatérési érték>
IS
    [<változók>]
BEGIN
    <függvénytörzs>
    RETURN <változó>;
    [<Exceptions>]
END;
/
```

- Meghívás: <függvéynév>(<paraméterek>)

DML pl/sql-ben

- Ugyanúgy mint sql-ben, de lehetőségünk van pl törölt sorok számát (SQL%ROWCOUNT) kiírni például.

Kurzor

- kurzor végigmegy egy reláció sorain
- Fetch: Kurzorhoz tartozó reláció következő sorának megállapítása

```
DECLARE CURSOR C_EMP IS
  (SELECT * FROM EMPLOYEES)
  R C_EMP%ROWTYPE;
BEGIN
  OPEN C_EMP;
  LOOP
    FETCH C_EMP INTO R;
    EXIT WHEN C_EMP%NOTFOUND;
    <kurzor-al valami számítása>
  END LOOP;
  CLOSE C_EMP;
END;
/
```

1 soros lekérdezés pl/sql-ben

- SELECT <lekérdezés> INTO <változók>
- Ha a lekérdezés sikeres, a változókba kerül az az 1 sor amit lekérdeztünk. Ezen változók elé rendre 1-1 kettőspontot kell írni.
- Ha 0, vagy 1-nél több sor a lekérdezés eredménye, akkor az SQLSTATE változóba kerül a megfelelő hibakód.

Funkcionális függőség

- Ha R két sora megegyezik az A1, A2, ..., An attribútumokon, akkor meg kell egyezniük más attribútumok egy B1, B2, ..., Bm sorozatán.

Szuperkulcs, illetve kulcs funkcionális

függőséggel

- Kulcs: Azt mondjuk, hogy az egy vagy több attribútumból álló $\{A_1, A_2 \dots A_n\}$ halmaz az R kulcsa, ha:
 1. Ezek az attribútumok funkcionálisan meghatározzák a reláció minden más attribútumát, azaz nem lehet R-ben két olyan különböző sor, amely mindegyik A_1, A_2, \dots, A_n -en megegyezne.
- Szuperkulcs: Minden szuperkulcs eleget tesz a kulcsdefiníció első feltételének: funkcionálisan meghatározzák a reláció összes többi attribútumát.

Armstrong axiómák

- Transzitivitás
- Reflexivitás
- Egyesítési szabály
- Szétvághatósági szabály
- Bővítési szabály

Egyesítési és szétvághatósági szabály

- SZ: Az $A_1, A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ függőséget helyettesíthetjük az $A_1 A_2 \dots A_n \rightarrow B_i$ ($i = 1, 2, \dots, m$) funkcionális függőségekből álló halmazzal.
- E: Az $A_1 A_2 \dots A_n \rightarrow B_i$ ($i = 1, 2, \dots, m$) funkcionális függőségekből álló halmazt helyettesíthetjük az $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ egyetlen függőséggel.

Függőségi halmaz minimális fedése

A minimális bázis egy relációhoz az a B bázis, amely az alábbi három feltételt elégíti ki:

1. B összes függőségének jobb oldalán egy attribútum van.
2. Ha bármelyik B-beli függőséget elhagyjuk, a fennmaradó halmaz már nem bázis.
3. Ha bármelyik B-beli funkcionális függőség bal oldaláról elhagyunk egy vagy több attribútumot, akkor az eredmény már nem marad bázis.

Nem veszteségmentes felbontás

- $R(A, B, C, D, E)$ egy felbontása: $R_1(A, B, C)$ és $R_2(C, D, E)$.

Függőségőrző felbontás

- $R = (A, B, C)$, $F = \{A \rightarrow B, B \rightarrow C\}$ R egy felbontása: $R_1 = (A, C)$ $R_2 = (B, C)$

BCNF

- Anomáliák megszüntetésére használjuk. Az R reláció BCNF-ben van akkor és csak akkor, ha minden olyan esetben, ha az R -ben érvényes egy $A_1A_2 \dots A_n \rightarrow B$ „nem triviális” függőség, akkor az $\{A_1, A_2, \dots, A_n\}$ halmaz R superkulcsa kell, hogy legyen. Függőség: „Ha R két sora megegyezik az A_1, A_2, \dots, A_n attribútumokon (azaz ezen attribútumok mindegyikéhez megfeleltetett komponensnek ugyanaz az értéke a két sorban), akkor meg kell egyezniük más attribútumok egy B_1, B_2, \dots, B_m sorozatán.” Superkulcs: Azokat az attribútum halmazokat, amelyek tartalmazznak kulcsot

Minden két-attribútumú reláció BCNF

- Az R reláció BCNF-ben van akkor és csak akkor, ha minden olyan esetben, ha R -ben érvényes egy $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ nem triviális függőség, akkor az $\{A_1, A_2, \dots, A_n\}$ halmaz R superkulcsa. Azaz minden nem triviális funkcionális függőség bal oldalának super kulcsnak kell lennie, ezért van minden két-attribútumú reláció BCNF-ben.

3NF

- A BCNF enyhébb alakja, amelyben előfordulhatnak olyan $X \rightarrow A$ funkcionális függőségek, ahol ha az X nem superkulcs, akkor az A valamely kulcsnak eleme.
- Az R reláció harmadik normálformában (3NF) van akkor és csak akkor, ha: valahányszor létezik R -ben egy $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ nem triviális függőség. Akkor vagy az $\{A_1, A_2, \dots, A_n\}$ halmaz az R superkulcsa, vagy azokra

az attribútumokra B_1, B_2, \dots, B_m közül, amelyek nincsenek az A -k között, teljesül, hogy egy kulcsnak az elemei (nem feltétlenül ugyanannak a kulcsnak).