

ADATBÁZISKEZELÉS (ST-PM)

Tematika

- 1 ✓ Az adatbázis-kezelés alapjai ✓
- 2 ✓ Adatbázis-rendszerek. Adatbázis adminisztrátor. ✓
- 3 ✓ Séma, alséma. Adat redundancia. ✓
- 4 ✓ Adatintegritás. Adatmodellek. Hálós-, hierarchikus-, relációs modellek. ✓
- 5 ✓ Egyed, egyedtípus ✓
- 6 ✓ Tulajdonság, tulajdonságtípus ✓
- 7 ✓ Kapcsolat. I:I, I:T, T:T kapcsolat leképezése. ✓
- 8 ✓ Kulcsok. Kulcsok a relációs modellben ✓
- 9 ✓ Egyed-kapcsolat diagrammok ✓
- 10 ✓ Funkcionális függőség. Tranzitivitás. ✓
- 11 ✓ Adatkezelési műveletek anomáliái ✓
- 12 ✓ Normalizálás. 1., 2., 3. normálformák. ✓

AB-G 10.

- FoxPro (VFP) fontosabb lehetőségei.
- 13 ✓ Parancsok, függvények
 - 14 ✓ Adatbázis és tábla kezelése, megnyitása. Munkaterületek használata. AK-G / 20 - 10 VFP (alapsz), 1 - 3
 - 15 ✓ Rekord-kezelés, index, reláció. AK-G/10 VFP 3 - 5
 - 16 ✓ Indexelés, keresés. Rendezés (fizikailag).
 - 17 ✓ Program kód létrehozása és futtatása. AB-G/14 -
 - 18 ✓ Függvények. Eljárások. AB-G / 15
 - 19 ✓ Beépített függvények. AB-G/30 - 34
 - 20 ✓ Memóriaváltozók, tömbök. AB-G/16 - 19 -
 - 21 ✓ Programozás: vezérlési szerkezetek. Ciklusok. ✓ AB-G/20 - 21 VFP: 14 - 16
 - 22 ✓ Form-ok. Vezérlések. Tulajdonságok. Események. Metodusok. ✓ VFP 7 - 18

SQL.

- 23 ✓ Lekérdezések ✓
- 24 ✓ Aggregáló függvények ✓
- 25 ✓ Csoportosító lekérdezések és rendezések ✓
- 26 ✓ Több táblára vonatkozó lekérdezések ✓
- 27 ✓ JOIN ... ON ... opció - Több tábla lek. ✓
- 28 ✓ Beágyazott lekérdezések ✓
- 29 ✓ Relációsémák definiálása ✓
- 30 ✓ Adatok módosítása az SQL-ben
- 31 ✓ Új sorok beszúrása ^{és} módosítás ✓
- 32 ✓ SELECT parancs a VFP-ban
- 33 ✓ Transzakciók (31) ✓
- 34 ✓ Nézet-táblák (VIEW-táblák) (32) ✓
- 35 ✓ KURSOR-ok FÜLET (33) ✓
- 36 ✓ Privilegiumok. GRANT és REVOKE parancsok (34-35) ✓
- ORACLE (alapok) (PM-szak !)

1. tétel: Az adatbázis-éselés alapjai. adatmodellek, adatbázisok. Hálós-, hierarchikus-, relációs adatbázisok, 1:1, 1:N, M:N kapcsolat leírása. Adatbázis-éselő rendszer.

- adat, információ

I. Adatbázisok:

- logikailag összetett info-és ömlesztve
- AB-nyelv : • adatbázisokat
- gyűjtés, tárolás és előfordulásait, szabványos eszköz } magába foglalja
- az értékek, értékek vagy nevek (ab-adminisztráció)

- séma, aléma
- metaadatok
- adatbázis, néző, rekord

II. AB-KR: (5hr.)

II. ANSI/SPARC modell:

ANSI: Amerikai Kongresszusi Hivatal
SPARC: Kongresszusi és Kormányzati Bizottsága

- 3 szint: - külső = felhasználói nézet
- középső = koncepcionális (logikai séma)
- belső = fizikai nézet

fizikai- és logikai adatfüggetlenség

IV. Adatmanipulációs nyelv: DDL

adatmanipulációs nyelv: DML

leérdeső nyelv: QL

index

IGL .& ., ., adatbázis, relációs adatbázisok és a relációs

V. Adatbiztonság: tranzakcióvezérlés, biztonsági másolatok, audit
zár, társasági joggal való megnyitás, ontt zár

VI. Adatmodellek:

konvenció:

- 1) info igény meghatározása
- 2) logikai ab-vezérlés
- 3) fizikai ab-előállítás

adatmodellek - egység /
- tulajdonság
- kapcsolat

Egység: egység típus
egység-előfordulás = rekord

Tulajdonság: tulajdonság típus időpont, időpont, időpont : időpont
-II- érték : 1 2 3 4 5 T

Érték, elsőleges érték

Kapcsolat: előfordulás, érték

1:1

1:M

M:N

VI. ~~Kisérlet~~ adatmodell típusai:

- hierarchikus
 - hálós
 - relációs
- 1968 IHS → első IBM által elkészített ARKA
relációs típus
SQL E.F. CODD 60-70-es évek

2. tétel: A relációs adatmodell kulcsok, funkcionális függőség.

- relációs adatmodell
- rel. adatmodell alapjai
- reláció (def + tábl.)
 - ↳ attribútum, előfordulás
 - rel. foka, rel. névszága
 - alapel., névmasztott rel (kézet)
- kapcsol. (1:1, 1:N, M:N)
 - ↳ kapcsolótábla
- funkcionális függőség
 - TRANZ.
 - REFL.
 - Armstrong-ax (egységsi, bővítési, névvalamintósági szabály)
 - F névűt vett lezáras
 - Eklésok (egyenlő, önkett, superklés, idegen eklés)

3. tétel: A normalizálás elmélete, gyakorlat. 1., 2., 3. normalformát.

- normalizálás
- adott névű megkeleő normalformájú
- 1, 2, 3 nf. Boyce-Codd nf.
- beszűrés, módosítás, törlési anomália
- relációalgebrai műveletek:
 - RUS
 - RNS
 - RJS
 - RXS
 - $\pi(R)$
 - $\sigma_f(R)$
 - $\rho_{\theta}(R)$

12. tétel: Bevezetés az SQL-be.

- SQL előnye:
- megkönnyíti megfogalmazni a matematikailag jól megala-
zott műveleteket
 - lehetőség van a programba történő beillesztésre
 - hatékony, gyors leírásos lehetőség biztosít
 - megkönnyíti a relációs jellegű
 - leírásos eredményei is relációk

- SQL parancs : parancs (paraméter) v. eredmény
↑
tábla, név, konstans
azonosító

SQL → lekérdező parancs
→ adatdefiniáló parancsok

13. tétel: Lekérdezések.

1) Egyszerű lekérdezések:

- SELECT (all, distinct) * <onlopnevlista> FROM <tábla név>
 - count
 - sum
 - avg
 - min
 - max
- ↑
aggregáló függvények
- <onlopnevlista> as <onlopnev>
 - LEFT (< karakter sorozat >, < darab >) (fgv)

2) Kialakító lekérdezések:

- where < feltétel >

↳ operátorok és operátorok
↓
konstans, rel. attr., onlopnev
fgv kihasználás

↳ összehasonlító, aritmetikai, logikai

- predikátum fgv:

- a) <onlopnevlista> BETWEEN <alsóérték> AND <felsőérték> ac < fc
- b) <ot.> [NOT] IN <értéklista> logikai eredmény
- c) <ot.> LIKE <"karakterlánc"> spec.: % , _

- YEAR (< dátum típus >) (fgv)

3) Összetett lekérdezések és rendezések:

GROUP BY <onlopnev>, ... dop.
HAVING <összetett feltétel >
ORDER BY <onlopnev v. onlopnevlista> (ASC/DESC), ...
↑
növekvő / csökkenő

Select 'A dolgozó nettó bére', AVG('A dolgozó fizetése') FROM dolgozo
GROUP BY 'A dolgozó nettó bére' HAVING ('A dolgozó fizetése') <= 120000.

Select 'A dolgozó neve' FROM dolgozo ORDER BY 'A dolgozó neve' (desc)

4) Több tábla-ra vonatkozó lekérdezések:

- select (all/distinct) * <onlopnevlista> FROM <tábla név> (<másodikonv>) ...
- <tábla név> CROSS JOIN <tábla név> Descartes - sorok

- $\langle \text{tábla}_1 \rangle \text{ JOIN } \langle \text{tábla}_2 \rangle \text{ ON } \langle \text{feltétel} \rangle$
- $\langle \text{tábla}_1 \rangle \text{ NATURAL JOIN } \langle \text{tábla}_2 \rangle$

"LÓGÓ" külső ömlesztés

- $\langle \text{tábla}_1 \rangle \text{ FULL OUTER JOIN } \langle \text{tábla}_2 \rangle \text{ ON } \langle \text{feltétel} \rangle$
- $\langle \text{tábla}_1 \rangle \text{ LEFT OUTER JOIN } \langle \text{tábla}_2 \rangle \text{ ON } \langle \text{feltétel} \rangle$
- $\langle \text{tábla}_1 \rangle \text{ RIGHT OUTER JOIN } \langle \text{tábla}_2 \rangle \text{ ON } \langle \text{feltétel} \rangle$
- $\langle \text{tábla}_1 \rangle \text{ NATURAL FULL OUTER JOIN } \langle \text{tábla}_2 \rangle$ sem ömlesztés
- $\langle \text{tábla}_1 \rangle \text{ NATURAL LEFT OUTER JOIN } \langle \text{tábla}_2 \rangle$
- $\langle \text{tábla}_1 \rangle \text{ NATURAL RIGHT OUTER JOIN } \langle \text{tábla}_2 \rangle$

... közbülső : lökés

5) Beágyazott lekérdezések:

a) egy értéket vizsgáló

pl.: $\text{in} \rightarrow \text{in}$ azokat, akik az állag alatt vannak.

b) egyenlőséget vizsgáló

- $\langle \text{konkretizálás} \rangle \text{ (NOT) IN } \langle \text{belső lekérdezés} \rangle$
- $\langle \text{konkretizálás} \rangle \langle \text{reláció} \rangle \text{ ALL | ANY } \langle \text{belső lekérdezés} \rangle$

Select * from dolgozó where 'A dolgozó fizetése' > ALL (select 'A dolgozó fizetése' from dolgozó where 'A dolgozó nem kegye' LIKE "Eger" OR 'A dolg. nem kegye' LIKE "Szeged").

c) általános relációt vizsgáló \rightarrow értéke feltétel:

- $\text{(NOT) EXISTS } \langle \text{belső lekérdezés} \rangle$ a lekérdezés rel. üres v. sem
- $\langle \text{konkretizálás} \rangle \dots \text{ (NOT) IN } \langle \text{belső lekérdezés} \rangle$

14. tétel: SQL parancsok:

1) Relációs adatbázis definíciója:

a) adattípusok

- egész számok: shortint, int, integer
- valós számok: float, real, decimal
- fix és változó hosszúságú kar. sorozatok: varchar(n), char(n), varchar, char
- dátum és idő: date, time

b) • CREATE TABLE $\langle \text{név} \rangle$ {attribútumdef.} ... }

attribútumdef: $\langle \text{név} \rangle \langle \text{típus} \rangle$ (default $\langle \text{érték} \rangle$)

- ALTER TABLE $\langle \text{név} \rangle$ ADD $\langle \text{attribútumdef} \rangle$
- " " $\langle \text{név} \rangle$ DROP $\langle \text{attribútumdef} \rangle$
- DROP $\langle \text{név} \rangle$

2) Változtatások az AB-ban:

- INSERT INTO $\langle \text{relációs név} \rangle$ VALUES $\langle \text{értéklista} \rangle$
- DELETE FROM $\langle \text{reláció} \rangle$ where $\langle \text{felt.} \rangle$
- UPDATE $\langle \text{reláció} \rangle$ set $\langle \text{értékadások} \rangle$ where $\langle \text{felt.} \rangle$

rules:	id's/keys	id's/keys
1 attr.	unique, primary key	reference
2 id's	primary key	foreign key

• reference (<id's> (<attr. id's>)) on delete (v. on update)
 set null / cascade.

not null
 check (<feltitel> } <attr. def>

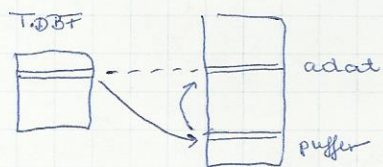
pl.: create assertion <id's> check (<feltitel>)

15. titel: SQL aralmarazda VFP-6au.

TRANZAKCIO (31)

- Commit work → változtatás végrehajtása } változóan
- rollback work → nem létező végrehajtást
- SET AUTOCOMMIT ON → tranzakció végén végrehajtja a változtatásokat
- SET AUTOCOMMIT OFF

begin transaction
rollback
end transaction



NÉZET (VIEW) TÁBLÁK (32)

- create view ujles-dial as select *

from students
where iegy = 5

- select * from ujles-dial;

- update stud2 set dial neve = 'balo' where studnum = 31;

- csoportosítás

- insert into only 5 values 31, 4

KURZOR (33)

{exec sql;} declare cursor cursor neve →

for select attributum →

from students → when orok = 1500 →

fetch : ideiglenes táblázat + sort sorok és a memóriában
tartó táblázatban.

elhelyezhető állásban

w = 10

count ... for ... to ...

for i = 1 to w

exec sql open cursor → megnyitás
exec sql close cursor → bezárás

ADATOK BIZTONSÁGA - PRIVILEGIUMOK (34)

• 13)

• Az adminisztrátor ad privilegiumokat

↓
select
insert, delete, update
references, index
synonym, alter
connect, resource, etc.

1. GRANT, REVOKE

pl: Grant → jog → on → tábla → to → Elnél →
grant → j → on → t → to → public →
grant → all → on → t → to → Elnél →
grant → j → on → t → to → E → with grant option

kontroll mód

I. tétel: Programozás: Veszélyes szerkesztés, átluszt.

1) if utasítás

```
if <felt>  
  <parancs>  
else  
  <parancs>  
endif.
```

pl: $y = \text{if}(x > 5, \text{"igaz"}, \text{"hamis"})$

```
if x > 5  
  y = "igaz"  
else  
  y = "hamis"  
endif
```

2) do case utasítás

```
do case  
  case <felt 1>  
    <parancs 1>  
  case <felt 2>  
    <parancs 2>  
  :  
  otherwise  
    <parancs(ok) w>  
endcase
```


Ciklusok:

1) do while < felt >
< parancs(ok) >
LOOP → visszaugrik a ciklus elejére
EXIT → kilép a ciklusból
enddo

2) for < ciklus-paraméter > = u1 to u3 (step u2)
< parancs(ok) >
(LOOP)
(EXIT)
endfor.

(step → növeli eggyel a reordmutatót.)

F. tétel. Programozás: algoritmusok, algoritmusok, algoritmusok