Nikovits Tibor
gyakorlata alapján

Adatbázisok tervezése, megvalósítása, menedzselése

(2008-2009/1)

FALUDI BENCE
(FABOAAI.ELTE)

E-mail: maxdamage870822@gmail.com
imacx@imacx.hu

Web: http://elte.imacx.hu (további jegyzetek)
http://www.imacx.hu (készítőről)

1.
Adatbázisok tervezése, megvalósítása, menedzselése

Adatbázis objektumok (DBA_OBJECTS)

Kinek a tulajdonában van a DBA_TABLES nevű nézet (illetve a DUAL nevű tábla)?

```
select owner from dba_objects where object_name = 'DBA_TABLES'
select owner from dba_tables where table_name = 'DUAL'
```

Kinek a tulajdonában van a DBA_TABLES nevű szinonima (illetve a DUAL nevű)?

```
select owner from dba_synonyms where synonym_name = 'DBA_TABLES'
select owner from dba_synonyms where synonym_name = 'DUAL'
```

Milyen típusú objektumai vannak az orauser nevű felhasználónak az adatbázisban?

```
select object_type from dba_objects where owner='ORAUSER'
```

Hány különböző típusú objektum van nyilvántartva az adatbázisban?

```
select count( unique object_type ) from dba_objects
```

Melyek ezek a típusok?

```
select unique object_type from dba_objects
```

Kik azok a felhasználók, akiknek több mint 10 féle objektumuk van?

```
select owner from dba_objects group by owner having count( unique object_type ) > 10
```

Kik azok a felhasználók, akiknek van triggere és nézete is?

```
select owner from dba_objects where object_type like '%TRIGGER%'
 union
select owner from dba_objects where object_type like '%VIEW%'
```

Kik azok a felhasználók, akiknek van nézete, de nincs triggere?

```
select owner from dba_objects where object_type like '%VIEW%'
 minus
select owner from dba_objects where object_type like '%TRIGGER%'
```

Kik azok a felhasználók, akiknek több mint 40 táblájuk, de maximum 37 indexük van?

```
select owner
from dba_objects
where object_type = 'INDEX'
group by owner, object_type
having count( object_type ) <= 37
 intersect
select owner
from dba_objects
where object_type = 'TABLE'
group by owner, object_type
having count( object_type ) > 40
```

Melyek azok az objektum típusok, amelyek tényleges tárolást igényelnek, vagyis tartoznak hozzájuk adatblokkok?

```
select distinct object_type from dba_objects where data_object_id is not null
```

Melyek azok az objektum típusok, amelyek nem igényelnek tényleges tárolást, vagyis nem tartoznak hozzájuk adatblokkok?

```
select distinct object_type from dba_objects where data_object_id is null
```

Táblák oszlopai (DBA_TAB_COLUMNS)

Hány oszlopa van a nikovits.emp táblának?

select * from dba_tab_columns where table_name = 'NIKOVITS.EMP'

Milyen típusú a nikovits.emp tábla 6. oszlopa?

select data_type from dba_tab_columns where column_id = 6 and table_name = NIKOVITS.EMP'

Adjuk meg azoknak a tábláknak a tulajdonosát és nevét, amelyeknek van 'Z' betűvel kezdődő oszlopa.

select owner, table_name from dba_tab_columns where column_name like 'Z%' group by table_name, owner

Adjuk meg azoknak a tábláknak a nevét, amelyeknek legalább 8 darab dátum típusú oszlopa van.

select table_name from dba_tab_columns group by data_type, table_name having data_type = 'DATE' and count(table_name) >= 8

Adjuk meg azoknak a tábláknak a nevét, amelyeknek 1. es 4. oszlopa is VARCHAR2 típusú.

```
select table_name
from dba_tab_columns
where data_type = 'VARCHAR2' and ( column_id = 1 or column_id = 4 )
group by table_name
having count( table_name ) = 2
```

Írjunk meg egy PLSQL procedúrát, amelyik a paraméterül kapott karakterlánc alapján kiírja azoknak a tábláknak a nevét és tulajdonosát, amelyek az adott karakterlánccal kezdődnek. (Ha a paraméter kisbetűs, akkor is működjön a procedúra!)

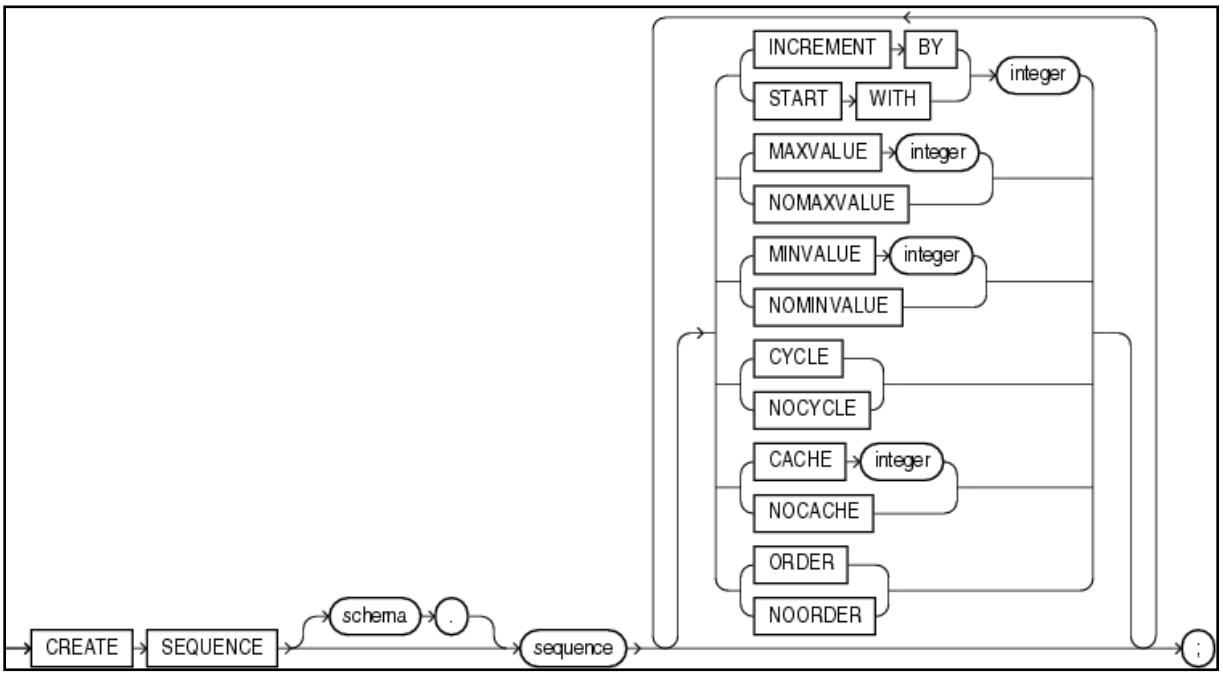
PROCEDURE tabla_kiiro(p_kar VARCHAR2)

A fenti procedúra segítségével írjuk ki a Z betűvel kezdődő táblák nevét és tulajdonosát.

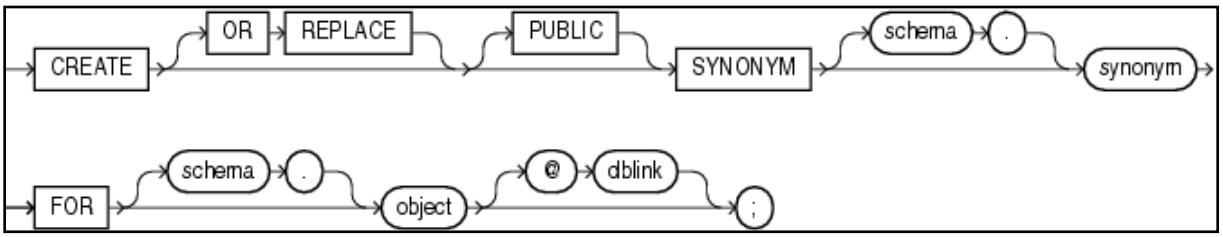
```
declare
 procedure getTablesLike( nev VARCHAR2 ) is
  rec dba_tables%rowtype;
 begin
  for rec in (select * from dba_tables where table_name like upper( nev ) || '%') loop
   dbms_output.put_line( rec.table_name || '   ' || rec.owner );
  end loop;
 end getTablesLike;
begin
 getTablesLike( 'z' );
end;
```

2.
Adatbázisok tervezése, megvalósítása, menedzselése

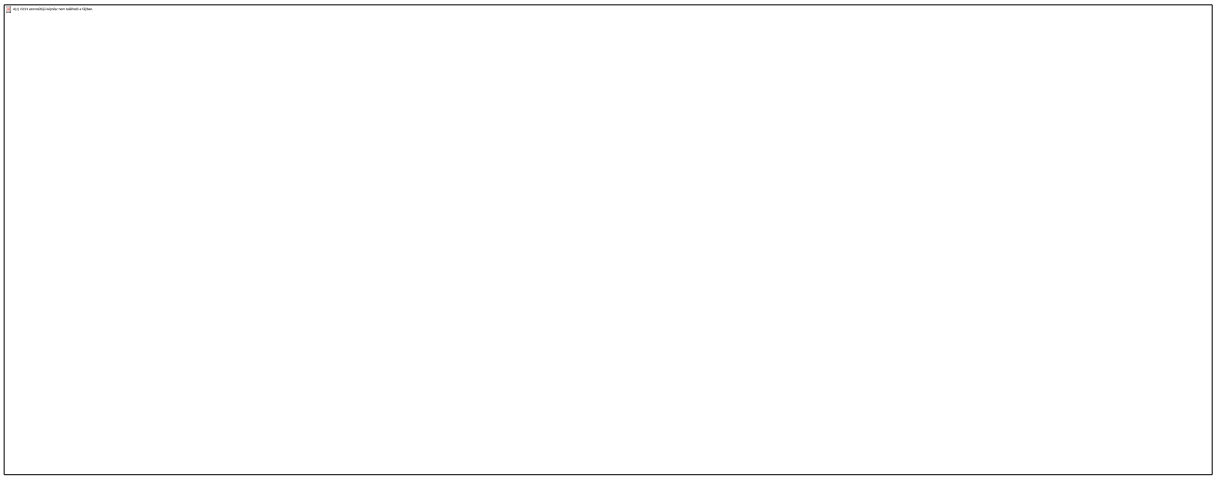
CREATE SEQUENCE



CREATE SYNONIM



CREATE DATABASE_LINK



Egyéb objektumok (szinonima, szekvencia, adatbázis-kapcsoló)
(DBA_SYNONYMS, DBA_SEQUENCES, DBA_DB_LINKS)

Szinonima létrehozása a NIKOVITS.EMP táblához

```
create or replace synonim szin1 for nikovits.emp
```

Adjuk ki az alábbi utasítást, SELECT * FROM szinonima1; majd derítsük ki, hogy kinek melyik tábláját kérdeztük le. (Ha esetleg nézettel találkozunk, azt is fejtsük ki, hogy az mit kérdez le.)

```
select table_name, table_owner from dba_synonyms where synonym_name = 'SZINONIMA1'
select * from dba_views where view_name = 'NEZET1'
```

Szekvencia létrehozása, majd attól új sorszám kérése, példa beszúrásra

```
create sequence s1              = új szekvencia létrehozása
select s1.nextval from dual     = következő érték kérése
select s1.currval from dual     = utolsó érték megadása
```

Hozzunk létre egy szekvenciát, amelyik az osztály azonosítókat fogja generálni a számunkra. Minden osztály azonosító a 10-nek többszöröse legyen. Vigyünk fel 3 új osztályt és osztályonként minimum 3 dolgozót a táblákba. Az osztály azonosítókat a szekvencia segítségével állítsuk elő, és ezt tegyük be a táblába. (Vagyis ne kézzel írjuk be a 10, 20, 30 ... stb. azonosítót.) A felvitel után módosítsuk a 10-es osztály azonosítóját a következő érvényes (generált) osztály azonosítóra. (Itt is a szekvencia segítségével adjuk meg, hogy mi lesz a következő azonosító.) A 10-es osztály dolgozóinak az osztályazonosító ertékét is módosítsuk az új értékre.

```
create sequence osztaly increment by 10 start with 10
create table dolgozo as select * from nikovits.dolgozo where dnev = ''
create table oszt as select * from nikovits.osztaly where oazon = 0

insert into oszt ( oazon, onev, telephely ) values ( osztaly.nextval, 'Osztaly ' || osztaly.currval,
'Telephely ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )

insert into oszt ( oazon, onev, telephely ) values ( osztaly.nextval, 'Osztaly ' || osztaly.currval,
'Telephely ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )

insert into oszt ( oazon, onev, telephely ) values ( osztaly.nextval, 'Osztaly ' || osztaly.currval,
'Telephely ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
insert into dolgozo ( dkod, dnev ) values ( osztaly.currval, 'Dolgozo ' || osztaly.currval )
```

Hozzatok létre adatbázis-kapcsolót (database link) az egyik adatbázisban, amelyik a másik adatbázisra mutat. Ennek segítségével adjátok meg a következő lekérdezéseket. A lekérdezések alapjául szolgáló táblák:

NIKOVITS.VILAG_ORSZAGAI ABLINUX adatbázis
NIKOVITS.FOLYOK ORADB adatbázis

```
create database link ablinux connect to faboaai identified by faboaai using 'ablinux'
```

Az országok egyedi azonosítója a TLD (Top Level Domain) oszlop. Az ország hivatalos nyelveit vesszőkkel elválasztva a NYELV oszlop tartalmazza. A GDP (Gross Domestic Product -> hazai bruttó össztermék) dollárban van megadva. A folyók egyedi azonosítója a NEV oszlop. A folyók vízhozama m3/s-ban van megadva, a vízgyűjtő területük km2-ben. A folyó által érintett országok azonosítóit (TLD) a forrástól a torkolatig (megfelelő sorrendben vesszőkkel elválasztva) az ORSZAGOK oszlop tartalmazza. A FORRAS_ORSZAG és TORKOLAT_ORSZAG hasonló módon a megfelelő országok azonosítóit tartalmazza. (Vigyázat!!! egy folyó torkolata országhatárra is eshet, pl. Duna)

Mely folyók érintik Csehországot?

```
select tld from nikovits.vilag_orszagai@ablinux where nev like '%Csehország%'
select * from nikovits.folyok where orszagok like '%cz%'

select * from nikovits.folyok where orszagok like '%' || ( select tld from
nikovits.vilag_orszagai@ablinux where nev like '%Csehország%' ) || '%'
```

Mely országokon folyik keresztül a Nílus? Az országokat a megfelelő sorrendben adjuk meg

- Házifeladat

Adattárolással kapcsolatos fogalmak

(DBA_TABLES, DBA_DATA_FILES, DBA_TEMP_FILES, DBA_TABLESPACES, DBA_SEGMENTS, DBA_EXTENTS, DBA_FREE_SPACE)

Adjuk meg az adatbázishoz tartozó adatfile-ok (és temporális fájlok) nevét és méretét méret szerint csökkenő sorrendben.

```
select file_name, bytes from dba_data_files
  union
select file_name, bytes from dba_temp_files
```

Adjuk meg, hogy milyen tablaterek vannak letrehozva az adatbazisban, az egyes tablaterek hany adatfajlbol allnak, es mekkora az osszmeretuk. (tablater_nev, fajlok_szama, osszmeret) !!! Vigyázat, van temporális táblatér is.

```
select tablespace_name, count( tablespace_name ), sum( bytes ) from dba_data_files group by
tablespace_name
  union
select tablespace_name, count( tablespace_name ), sum( bytes ) from dba_temp_files group by
tablespace_name
```

Mekkora a blokkok merete a USERS táblatéren?

```
select tablespace_name, block_size from dba_tablespaces
```

Melyik a legnagyobb méretű tábla szegmens az adatbázisban (a tulajdonost is adjuk meg) és hány extensből áll? (A particionalt tablakat most ne vegyuk figyelembe.)

```
select * from (select owner, extents, bytes from dba_segments where segment_type = 'TABLE' order
by bytes desc) where rownum = 1
```

Melyik a legnagyobb meretű index szegmens az adatbazisban és hány blokkból áll? (A particionalt indexeket most ne vegyuk figyelembe.)

```
select * from (select segment_name, blocks, bytes from dba_segments where segment_type = 'INDEX'
order by bytes desc) where rownum = 1
```

Adjuk meg adatfájlonként, hogy az egyes adatfájlokban mennyi a foglalt hely összesen (írassuk ki a fájlok méretét is).

```
select D.file_name, sum( E.bytes ), D.bytes
from dba_extents E, dba_data_files D
where D.file_id = E.file_id
group by D.file_name, D.bytes
```

Melyik két felhasználó objektumai foglalnak összesen a legtöbb helyet az adatbázisban?

```
select * from (select owner, sum( bytes )
   from dba_extents
   group by owner
   order by sum( bytes ) desc)
where rownum <= 2
```

Van-e a NIKOVITS felhasználónak olyan táblája, amelyik több adatfájlban is foglal helyet?

```
select segment_name, count( distinct file_id)
from dba_extents
where owner = 'NIKOVITS' and
    segment_type = 'TABLE'
group by segment_name
having count( distinct file_id ) > 1;
```

Melyik táblatéren van az ORAUSER felhasználó dolgozó táblája?

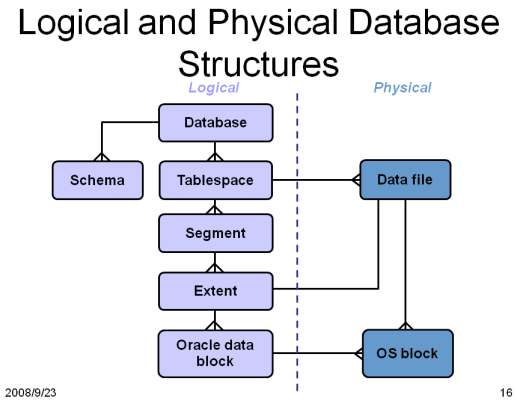
```
select tablespace_name
from dba_tables
where owner = 'ORAUSER' and table_name = 'DOLGOZO'
```

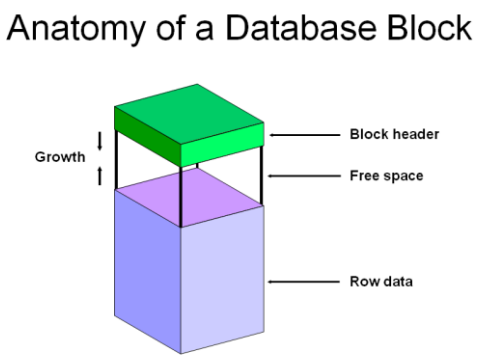
Melyik táblatéren van a NIKOVITS felhasználó ELADASOK táblája? (Miért lesz null?)

```
select tablespace_name
from dba_tables
where owner = 'NIKOVITS' and table_name = 'ELADASOK'
```

3.
Adatbázisok tervezése, megvalósítása, menedzselése

Elméleti alapozás





ROWID adattípus formátuma és jelentése (lásd még DBMS_ROWID package)

18 karakteren irodik ki, a kovetkezo formaban: OOOOOOFFFBBBBBBRRR

OOOOOO	az objektum azonositoja
FFF	fajl azonositoja (tablateren beluli relativ sorszam)
BBBBBB	blokk azonosito (a fajlon beluli sorszam)
RRR	sor azonosito (a blokkon beluli sorszam)

A ROWID megjeleniteskor 64-es alapu kodolasban jelenik meg. Az egyes szamoknak (0-63) a következo karakterek felelnek meg: A-Z -> (0-25), a-z -> (26-51), 0-9 -> (52-61), '+' -> (62), '/' -> (63)
Pl. 'AAAAAB' -> 000001

Könyvajánló:
Molina, Ullman, Widom: Adatbázisrendszerek megvalósítása

Adattárolással kapcsolatos fogalmak
(DBA_TABLES, DBA_DATA_FILES, DBA_TEMP_FILES, DBA_TABLESPACES, DBA_SEGMENTS, DBA_EXTENTS, DBA_FREE_SPACE)

Van-e a NIKOVITS felhasználónak olyan táblája, amelyik több adatfájlban is foglal helyet?

```
select extent_id, count( distinct file_id )
from dba_extents
where owner = 'NIKOVITS' and segment_type = 'TABLE'
having count( distinct file_id ) > 1
group by extent_id
```

Melyik táblatéren van az ORAUSER felhasználó dolgozo táblája?

```
select segment_name, tablespace_name
from dba_segments
where owner = 'ORAUSER' and segment_name = 'DOLGOZO' and segment_type = 'TABLE'
```

Melyik táblatéren van a NIKOVITS felhasználó ELADASOK táblája? (Miért lesz null?)

```
select segment_name, segment_type, tablespace_name
from dba_segments
```

```
where owner = 'NIKOVITS' and segment_name = 'ELADASOK';

select tablespace_name, table_name
from dba_tables
where owner = 'NIKOVITS' and table_name = 'ELADASOK';
```

Más táblatéren van a tábla, mivel particionált tábláról van szó.

ROWID adattípus formátuma és jelentése (lásd még DBMS_ROWID package)

A NIKOVITS felhasználó CIKK táblája hány blokkot foglal le az adatbázisban? (Vagyis hány olyan blokk van, ami ehhez a táblához van rendelve és így azok már más táblákhoz nem adhatók hozzá?)

```
select segment_name, blocks
from dba_extents
where owner = 'NIKOVITS' and segment_name = 'CIKK' and segment_type = 'TABLE';

-- select cnev, rowid from nikovits.cikk;
-- select * from nikovits.cikk where rowid = 'AAAUDMAAEAAAAJKAAA';
```

A NIKOVITS felhasználó CIKK táblájának adatai hány blokkban helyezkednek el? (Vagyis a tábla sorai ténylegesen hány blokkban vannak tárolva?) !!! -> Ez a kérdés nem ugyanaz, mint az előző.

```
select count( distinct substr( rowid, 1, 15 ) )
from nikovits.cikk
```

Az egyes blokkokban hány sor van?

```
select substr(rowid,1,15), count( distinct (substr( rowid, 16, 3 ) ) ) from nikovits.cikk group by substr(
rowid, 1, 15 )
```

Hozzunk létre egy táblát az EXAMPLE táblatéren, amelynek szerkezete azonos a nikovits.cikk táblával és pontosan 128 KB helyet foglal az adatbázisban. Foglaljunk le manuálisan egy újabb 128 KB-os extenst a táblához. Vigyünk fel sorokat addig, amíg az első blokk tele nem lesz, és 1 további sora lesz még a táblának a második blokkban. (A felvitelt PLSQL programmal végezzük és ne kézzel, mert úgy kicsit sokáig tartana.)

```
create table cikk
tablespace EXAMPLE
storage (initial 128K)
 as
select * from nikovits.cikk where 0 = 1

select segment_name, bytes
from dba_extents
where owner = 'FABOAAI' and segment_name = 'CIKK' and segment_type = 'TABLE';

alter table cikk allocate extent (size 128K)
```

- PLSQL ciklus házifeladat

Állapítsuk meg, hogy a SH.SALES táblának a következő adatokkal azonosított sora (time_id='1999.04.10', prod_id=2860, cust_id=37280) melyik adatfájlban van, azon belül melyik blokkban, és a blokkon belül hányadik a sor?

```
select dbms_rowid.rowid_relative_fno(rowid), dbms_rowid.rowid_block_number(rowid),
dbms_rowid.rowid_row_number(rowid)
from sh.sales
where time_id = to_date( '1999.04.10', 'YYYY.MM.DD' ) and prod_id=2860 and cust_id=37280
```

Az előző feladatban megadott sor melyik partícióban van? (* biztos rossz)

```
select distinct partition_name from dba_tab_partitions
where table_name = (
 select table_name from dba_tables
 where
  table_name = (
   select object_name from dba_objects
   where
    object_id = (
     select dbms_rowid.rowid_object(rowid)
     from sh.sales
     where time_id = to_date( '1999.04.10', 'YYYY.MM.DD' ) and prod_id=2860 and cust_id=37280 )
   ) and
  owner = (
   select owner from dba_objects
   where
    object_id = (
     select dbms_rowid.rowid_object(rowid)
     from sh.sales
     where time_id = to_date( '1999.04.10', 'YYYY.MM.DD' ) and prod_id=2860 and cust_id=37280)
   )
  )
```

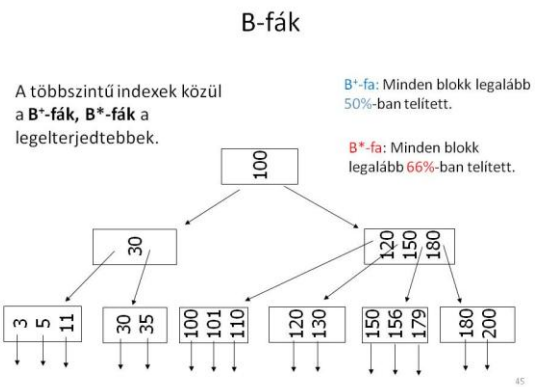
Mennyi az objektum azonosítója, és ez milyen objektum?

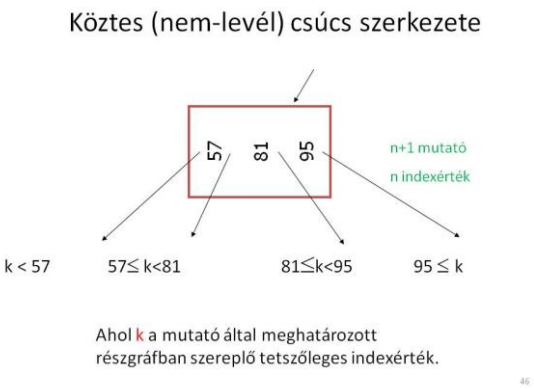
```
select object_id, object_type from dba_objects
where
 object_id = (
  select dbms_rowid.rowid_object(rowid)
  from sh.sales
  where time_id = to_date( '1999.04.10', 'YYYY.MM.DD' ) and prod_id=2860 and cust_id=37280 )
```

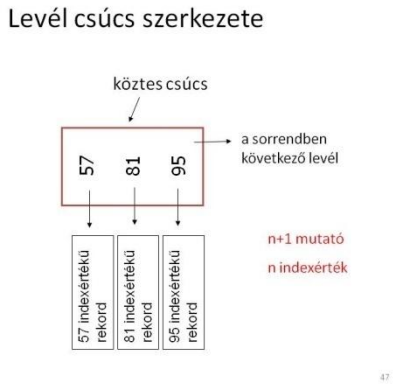
4.
Adatbázisok tervezése, megvalósítása, menedzselése

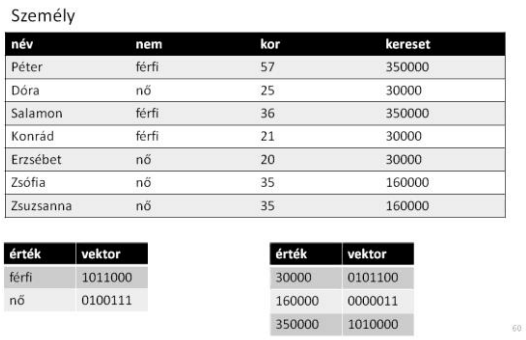
Elméleti alapozás

B-fával kapcsolatos anyagok: http://people.inf.elte.hu/balhal/AB%20terv%20menedzs%20megval/fajlszerv.ppt









Tömörítés (Bitvektorok szakaszhossz kódolása)

Ha a táblában n rekord van, a vizsgált attribútum pedig m különböző értéket vehet fel, ekkor, ha m nagy, a bitmap index túl naggyá is nőhet (n*m méret). Ebben az esetben viszont a bitmap indexben az egyes értékekhez tartozó rekordokban kevés az 1-es. A tömörítési technikák általában csak ezeknek az 1-eseknek a helyét határozzák meg.

Tegyük fel, hogy i db 0-t követ egy 1-es. Legegyszerűbb megoldásnak tűnik, ha i-t binárisan kódoljuk. Ám ez a megoldás még nem jó: (a 000101 és 010001 vektorok kódolása is 111 lenne).

Tegyük fel, hogy i binárisan ábrázolva j bitből áll. Ekkor először írjunk le j-1 db 1-est, majd egy 0-t, és csak ez után i bináris kódolását.

Példa: a 000101 kódolása: 101001, a 010001 kódolása: 011010.

Indexek

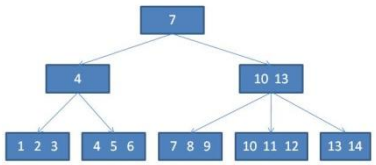
CREATE INDEX I1 ON T (O1,O2,O3 DESC): DBA_INDEXES-ben az index 1 sor, és ez tartalmaz dolgokat, míg a DBA_IND_COLUMNS pedig oszloponként tárolja az adatokat.

Indexek

(DBA_INDEXES, DBA_IND_COLUMNS, DBA_IND_EXPRESSIONS)

DKOD	DNEV	FIZETES	FOGLALKOZAS	OAZON
1	SMITH	800	CLERK	20
2	ALLEN	1600	SALESMAN	30
3	WARD	1250	SALESMAN	30
4	JONES	2975	MANAGER	20
5	MARTIN	1250	SALESMAN	30
6	BLAKE	2850	MANAGER	30
7	CLARK	2450	MANAGER	10
8	SCOTT	3000	ANALYST	20
9	KING	5000	PRESIDENT	10
10	TURNER	1500	SALESMAN	30
11	ADAMS	1100	CLERK	20
12	JAMES	950	CLERK	30
13	FORD	3000	ANALYST	20
14	MILLER	1300	CLERK	10

Készítsen B-fa indexet a dolgozó tábla DKOD oszlopára. Tegyük fel, hogy egy blokkba 3 bejegyzés fér el. Rajzolja le a fát.



Készítsen bitmap indexet a dolgozó tábla OAZON oszlopára. Adja meg a bitvektorokat és a felépített B-fát is.

10: 00000010100001

6 1 4

Tömörítse a kapott bitvektorokat a szakaszhossz kódolással.

10: 11011001110100

Fejtsük vissza a következő, szakaszhossz kódolással tömörített bitvektort: 11101101001011

00000000000000110001

Hozzunk létre egy vagy több táblához több különböző indexet, legyen köztük több oszlopos, csökkenő sorrendű, bitmap, függvény alapú stb. (Ehhez használhatjátok az előadáson elhangzottakat, és az ott szereplő példákat.) Az alábbi lekérdezésekkel megállapítjuk az iménti indexeknek mindenféle tulajdonságait a katalógusokból.

```
create index ind1 on cikk (ckod)
create index ind2 on dolgozo (dkod desc)
create index ind3 on oszt (oazon)
```

Adjuk meg azoknak a tábláknak a nevét, amelyeknek van csökkenő sorrendben indexelt oszlopa.

```
select * from dba_ind_columns where descend = 'DESC'
```

Adjuk meg azoknak az indexeknek a nevét, amelyek legalább 9 oszloposak. (Vagyis a táblának legalább 9 oszlopát vagy egyéb kifejezését indexelik.)

```
select index_name
from dba_ind_columns
having count( table_name ) >= 9
group by table_name, index_name, index_owner
```

Adjuk meg az SH.SALES táblára létrehozott bitmap indexek nevét.

```
select index_name
from dba_indexes
where index_type = 'BITMAP' and owner = 'SH' and table_name = 'SALES'
```

Adjuk meg azon kétoszlopos indexek nevét és tulajdonosát, amelyeknek legalább az egyik kifejezése függvény alapú.

```
select index_name, index_owner
from dba_ind_columns
having count( table_name ) >= 2
group by table_name, index_name, index_owner
 intersect
select index_name, owner
from dba_indexes
where index_type like '%FUNCTION-BASED%'
```

Adjuk meg az egyikükre, pl. az OE tulajdonában lévőre, hogy milyen kifejezések szerint vannak indexelve a soraik. (Vagyis mi a függveny, ami alapján a bejegyzések készülnek.)

```
select column_expression from dba_ind_expressions
where index_owner = 'OE' and
    index_name =
     (select index_name
      from dba_ind_columns
      having count( table_name ) >= 2
      group by table_name, index_name, index_owner
       intersect
      select index_name
      from dba_indexes
      where index_type like '%FUNCTION-BASED%' and
          owner = 'OE')
```

Adjuk meg a NIKOVITS felhasználó tulajdonában levő index-szervezett táblák nevét.

```
select * from dba_tables where iot_type like 'IOT%' and owner = 'NIKOVITS'
```

Adjuk meg a fenti táblák index részét, és azt, hogy ezek az index részek (szegmensek) melyik táblatéren vannak?

```
select * from dba_indexes where owner=”NIKOVITS’ and table_name is (select * from dba_tables
where iot_type like 'IOT%' and owner = 'NIKOVITS'
```

Keressük meg a szegmensek között az előző táblákat illetve indexeket, és adjuk meg a méretüket.

```
select * from dba_segments where owner = ’NIKOVITS’ and segment_name in ( select index_name
from dba_indexes where owner=’NIKOVITS’ and table_name = ’CIKK_IOT’ )
```

Keressük meg az adatbázis objektumok között a fenti táblákat és indexeket, és adjuk meg az objektum azonosítójukat és adatobjektum azonosítójukat (DATA_OBJECT_ID).

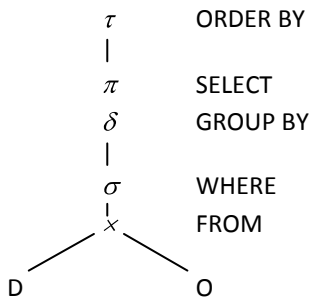
```
select * from dba_objects where owner = ’NIKOVITS’ and object_name in ( select index_name from
dba_indexes where owner=’NIKOVITS’ and table_name = ’CIKK_IOT’ )

select * from dba_objects where owner = ’NIKOVITS’ and object_name = ’CIKK_IOT’
```

5.
Adatbázisok tervezése, megvalósítása, menedzselése

ZH: 2 hét múlva lehet rá számítani, lesz benne PSSQL, SQL, és papíron (segédlet nélküli) elmélet számonkérés.

Kiterjesztett relációs algebrai műveletek

Jelölések és értelmezésük multi halmazok esetén:



Unió: $R \cup S$
(előfordulások **összege**)

Metszet: $R \cap S$
(előfordulások **minimuma**)

Különbség: $R - S$
(előfordulások **különbsége**)

Kiválasztás: $\sigma_c(R)$
(ahol **C** egy feltétel, AND, OR, NOT megengedett)

Vetítés: $\pi_L(R)$
(L-ben **E->z** megengedett, ahol E egy kifejezés, és z ennek az új neve)

Szorzat: $R \times S$

Ismétlődések kiküszöbölése: $\delta(R)$

Csoportosítás és összesítés: $\gamma_i(R)$
(L-ben attribútumok és összesítő fv-ek lehetnek valamint ezek átnevezése)

Rendezés: $\tau_L(R)$
(L-ben attribútumok listája szerepelhet)
(algebrai kifejezésben csak utolsó operátorként van értelme beszélni a τ operátorról)

Példa:

```
SELECT onev, AVG(fizetes) + 100 emelt
FROM dolgozo, osztaly
WHERE dolgozo.oazon=osztaly.oazon
GROUP BY onev
HAVING COUNT(dkod) > 3
ORDER BY onev;
```

$\tau_{onev}(\pi_{onev,avg+100->emelt}(\sigma_{cnt>3}(\gamma_{onev,AVG(fizetes)->avg,COUNT(dkod)->cnt}(\sigma_{D.oazon=O.oazon}(D \times O)))))$

Az operátorok egymás utáni alkalmazását egy **kifejezésfa** formájában rajzolhatjuk fel.

Feladatok

DKOD	DNEV	FIZETES	FOGLALKOZAS	BELEPES	OAZON
1	SMITH	800	CLERK	1980	20
2	ALLEN	1600	SALESMAN	1981	30
3	WARD	1250	SALESMAN	1981	30
4	JONES	2975	MANAGER	1981	20
5	MARTIN	1250	SALESMAN	1981	30
6	BLAKE	2850	MANAGER	1981	30
7	CLARK	2450	MANAGER	1981	10
8	SCOTT	3000	ANALYST	1982	20
9	KING	5000	PRESIDENT	1981	10
10	TURNER	1500	SALESMAN	1981	30
11	ADAMS	1100	CLERK	1983	20
12	JAMES	950	CLERK	1981	30
13	FORD	3000	ANALYST	1981	20
14	MILLER	1300	CLERK	1982	10

Tegyük fel, hogy a FOGLALKOZAS, a BELEPES és az OAZON oszlopokra létezik bitmap index (3 index). Készítsük el az alábbi lekérdezésekhez szükséges bitvektorokat, majd végezzük el rajtuk a szükséges műveleteket, és adjuk meg azt az előállt bitvektort, ami alapján a végeredmény sorok megkaphatók. Ellenőrzésképpen adjuk meg a lekérdezést SQL-ben is.

FOGLALKOZAS:	**CLERK:**	10000000001101
	SALESMAN:	01101000010000
	MANAGER:	00010110000000
	ANALYST:	00000001000010
	PRESIDENT:	00000000100000
BELEPES:	**1980:**	10000000000000
	1981:	01111110110110
	1982:	00000001000001
	1983:	00000000001000
OAZON:	**10:**	00000010100001
	20:	10010001001010
	30:	01101100010100

- **Adjuk meg azoknak a dolgozóknak a nevét, akik 1981-ben léptek be és a foglalkozásuk hivatalnok (CLERK), vagy a 20-as osztályon dolgoznak és a foglalkozásuk MANAGER.**

```
select dnev from dolgozo where belepes = to_date( 'YYYY', '1981' ) and foglalkozas = 'CLERK'
  union
select dnev from dolgozo where oazon = 20 and foglalkozas = 'MANAGER'
```

$$\begin{array}{r} 01111110110110 \\ \textbf{and}\ 10000000001101 \\ 00000000000100 \end{array} \qquad \begin{array}{r} 10010001001010 \\ \textbf{and}\ 00010110000000 \\ 00010000000000 \end{array}$$

$$\begin{array}{r} 00000000000100 \\ \textbf{or}\ 00010000000000 \\ 00010000000100 \end{array}$$

A 4. és a 12. sorszámú személy megfelel a feltételeknek: JONES, JAMES

- **Adjuk meg azoknak a dolgozóknak a nevét, akik nem 1981-ben léptek be és a 10-es vagy a 30-as osztályon dolgoznak.**

```
select dnev from dolgozo where belepes is not to_date( 'YYYYY', '1981' )
  intersect
select dnev from dolgozo where oazon in (10, 30)
```

$$\begin{matrix} \textbf{\textit{not}} & 01111110110110 \\ & 10000001001001 \end{matrix} \qquad \begin{matrix} \textbf{\textit{or}} & 00000010100001 \\ & 01101100010100 \\ & 01101110110101 \end{matrix}$$

$$\begin{matrix} \textbf{\textit{and}} & 10000001001001 \\ & 01101110110101 \\ & 00000000000001 \end{matrix}$$

A 14. sorszámú személy megfelel a feltételeknek: MILLER

Adjuk, meg a kiterjesztett relációs algebrai kifejezését az alábbi lekérdezéseknek majd rajzoljuk fel a kifejezésfát is.

- **Adjuk meg osztályonként az osztály nevét és az ott dolgozók számát a dolgozók száma szerint növekvő sorrendben.**

$$\pi_{fő}\left(\gamma_{onev,count(dkod)\to fő}\left(\sigma_{D.oazon=O.oazon}(D \times O)\right)\right)$$

$$\pi \to \gamma \to \sigma \to \times < \begin{matrix} D \\ O \end{matrix}$$

- **Adjuk meg azoknak az osztályoknak a nevét, ahol az átlagfizetés nagyobb, mint 2000.**

$$\pi_{O.onev}\left(\sigma_{count(D.fizetes)>2000}\left(\gamma_{D.oazon,O.onev}\left(\sigma_{O.oazon=D.oazon}(D \times O)\right)\right)\right)$$

$$\pi \to \sigma \to \gamma \to \sigma < \begin{matrix} D \\ O \end{matrix}$$

- **Adjuk meg azoknak a foglalkozásoknak a nevét, amelyek a 10-es és 20-as osztályon is előfordulnak. Ismétlődések ne legyenek a végeredményben.**

$$\delta\left(\pi_{foglalkozás}\left(\sigma_{oazon=10}(D)\right) \cap \pi_{foglalkozás}\left(\sigma_{oazon=20}(D)\right)\right)$$

$$\delta \to \cap < \begin{matrix} \pi \to \sigma \to D \\ \pi \to \sigma \to D \end{matrix}$$

Tegyük fel, hogy a dolgozó tábla sorai egyenként 1 blokkot foglalnak el, és a memóriánk 4 blokknyi. Rendezzük a tábla sorait fizetés szerint egy rendezés alapú algoritmussal. Adjuk meg az első menet után a rendezett részlistákat (elég a dnev, fizetest megadni). Hány menetes algoritmusra lesz szükségünk?

$$M = 4$$

1: 800, 1250, 1600, 2975 **2:** 1250, 2450, 2850, 3000 …

Tegyük most fel, hogy a memóriánk 6 blokknyi és van még egy vásárlás tábla, aminek a szerkezete a következő: VASARLAS(dkod, cikk, mennyiseg, ar). Ennek a táblának a sorai is 1 blokkot foglalnak és a tábla kb. 120 sorból áll. Mennyi a műveletigénye egy hash alapú, egy rendezés alapú és egy beágyazott ciklusos algoritmusnak, ami arra válaszol, hogy az egyes dolgozók összesen mennyit költöttek? Feltehetjük, hogy az összegeket gyűjtő számlálók még befférnek a memóriába a blokkok mellett. Írjuk le röviden, hogy az egyes algoritmusok hogyan fognak működni. Adjuk meg a kosarakat a hasítás alapú algoritmus első menete után.

$$M = 6$$
$$Vásárlás\colon 120\ sor \to 1\ blokk/sor$$

$$Dolgozó\colon 14\ sor \rightarrow 1\ blokk/sor$$

Nasted Loop: $1 * Dolgozó + 3 * Vásárlás = 14 * 3 * 120 = 374$ (ha a kisebbel kezdek)
$24 * Dolgozó + 1 * Vásárlás = 24 * 14 + 120 = 456$ (ha a nagyobbal kezdjük)

5 blokkonként beolvassuk a dolgozó tábla elemeit, és hozzá az egész Vásárlás táblát, így először 5, 5, 4-et olvasunk, ami 3 menet, azaz 3x olvassuk be a Vásárlás táblát, és egyszer megyünk végig a Dolgozó táblán.

Hash Join: $3 * Dolgozó + 3 * Vásárlás = 3 * 13 + 3 * 120 = 402$

Először beolvassuk a Dolgozót és kihasítjuk az outputra, majd beolvassuk a Vásárlót és kihasítjuk az outputra, majd beolvassuk a végeredményt és ezt feldolgozzuk, ezzel mégegy olvasást produkálva.

Sort: $5 * Dolgozó + 5 * Vásárlás = 5 * 14 + 5 * 120 = 670$

Szétbontom 6-osával, tehát 5,5,4 a Dolgozó tábla esetén, míg 20 ilyen csoport a Vásárlás tábla esetén. ezután ezeket összefuttatjuk, a Dolgozókat egy ilyen csoportba rendezve, míg

Partícionálás

(DBA_PART_TABLES, DBA_PART_INDEXES, DBA_TAB_PARTITIONS, DBA_IND_PARTITIONS, DBA_TAB_SUBPARTITIONS, DBA_IND_SUBPARTITIONS, DBA_PART_KEY_COLUMNS)

Adjuk meg az SH felhasználó tulajdonában levő partícionált táblák nevét és a particionálás típusát.

```
select table_name, partitioning_type from dba_part_tables where owner = 'SH'
```

Soroljuk fel az SH.COSTS tábla partícióit valamint, hogy hány blokkot foglalnak az egyes partíciók. (Vigyázat! Különböző értéket kaphatunk a különböző adatszótárakban. Ennek magyarázatát lásd később az ANALYZE parancsnál)

```
select partition_name, blocks
from dba_tab_partitions
where table_owner = 'SH' and table_name = 'COSTS'

select partition_name, blocks
from dba_segments
where owner = 'SH' and
    segment_type = 'TABLE PARTITION' and
    partition_name IN (
      select partition_name
      from dba_tab_partitions
      where table_owner = 'SH' and table_name = 'COSTS' )
```

Adjuk meg, hogy az SH.COSTS tábla mely oszlop(ok) szerint van particionálva.

```
select column_name
from dba_part_key_columns
where owner = 'SH' and name = 'COSTS'
```

Adjuk meg, hogy a NIKOVITS.ELADASOK3 illetve az SH.COSTS táblák második partíciójában milyen értékek szerepelhetnek.

```
select *
from dba_tab_partitions
where table_owner = 'NIKOVITS' and
    table_name like 'ELADASOK%'
```

Adjuk meg egy partícionált tábla logikai és fizikai részeit (pl. NIKOVITS.ELADASOK). Maga a tábla most is logikai objektum, a partíciói vannak fizikailag tárolva. Nézzük meg az objektumok és a szegmensek között is.

```
select * from dba_objects
where owner = 'NIKOVITS' and
```

```
    object_name IN (
      select table_name
      from dba_tab_partitions
      where table_owner = 'NIKOVITS' and table_name = 'ELADASOK' )

// Ahol TABLE van az logikai, ahol TABLE_PARTITION az pedig fizikai

select * from dba_segments
where owner = 'NIKOVITS' and
    segment_name IN (
      select table_name
      from dba_tab_partitions
      where table_owner = 'NIKOVITS' and table_name = 'ELADASOK' )

// Itt már csak 3-at látunk, mert a logikaihoz nem tartozik szegmens
```

Illetve ha alpartíciói is vannak (pl. nikovits.eladasok4), akkor csak az alpartíciók vannak tárolva. Nézzük meg az objektumok és a szegmensek között is.

```
select * from dba_objects
where owner = 'NIKOVITS' and
    object_name = 'ELADASOK4'
```

Melyik a legnagyobb méretű partícionált tábla az adatbázisban a partíciók összméretét tekintve? (az alpartícióval rendelkező táblákat is vegyük figyelembe)

```
select * from
(
 select partition_name, sum( blocks ) from dba_segments
 where partition_name is not null and
      segment_type in ( 'TABLE PARTITION', 'TABLE SUBPARTITION' )
 group by partition_name
 order by sum( blocks ) desc
)
where rownum < 2
```

A SZALLIT_PART tábla egyik sora (pl.: ckod=14, pkod=824, szkod=20) melyik adat-objektumban található?

```
select data_object_id from dba_objects
where object_id = (
select dbms_rowid.rowid_object(rowid)
from nikovits.szallit
where szkod = 14 and ckod = 824 and pkod = 20 )
```

6.
Adatbázisok tervezése, megvalósítása, menedzselése

Első zárthelyi

- Papíron feladatok megoldása (nem használható segédanyag)
  - Bitindexek kiszámítása
  - JOIN költségek kiszámolása
  - Fastruktúra felrajzolása
  - Elméleti kérdések: „*mi a különbség a sűrű és ritka index között?*”, „milyen szempontok lehetségesek a partícionálás esetén? – Intervallum, Hash, Lista, vagy ezek keverékei”

- Gép melletti feladatok (jegyzetek/példaprogramok használhatóak)
  - PLSQL: paraméterek alapján valamilyen információ visszaadása
  - SQL: hasonló feladatok, mint a gyakorlaton. (javasolt átnézni, hogy a méreteket hogy tárolja)

Klaszter (CLUSTER)
(DBA_CLUSTERS, DBA_CLU_COLUMNS, DBA_TABLES, DBA_CLUSTER_HASH_EXPRESSIONS)

Hozzunk létre egy DOLGOZO(dazon, nev, beosztas, fonoke, fizetes, oazon ... stb.) és egy OSZTALY(oazon, nev, telephely ... stb.) nevű táblát. (lásd NIKOVITS.DOLGOZO és NIKOVITS.OSZTALY) A két táblának az osztály azonosítója (oazon) lesz a közös oszlopa. A két táblát egy index alapú CLUSTEREN hozzuk létre. (Előbb persze létre kell hozni a clustert is.) Majd tegyünk bele 3 osztályt, és osztályonként két dolgozót.

```
create cluster klaszter
 ( oazon NUMBER(2) );

create index idx_oazon on cluster klaszter;

create table osztaly
cluster klaszter ( oazon )
as
 select * from nikovits.osztaly

create table dolgozo
cluster klaszter ( oazon )
as
 select * from nikovits.dolgozo
```

Adjunk meg egy olyan clustert az adatbázisban (ha van ilyen), amelyen még nincs egy tábla sem.

```
SELECT owner, cluster_name FROM dba_clusters
 MINUS
SELECT owner, cluster_name FROM dba_tables;
```

Adjunk meg egy olyant, amelyiken legalább 6 darab tábla van.

```
SELECT owner, cluster_name FROM dba_tables WHERE cluster_name IS NOT NULL
GROUP BY owner, cluster_name HAVING COUNT(*) >= 6;
```

Adjunk meg egy olyan clustert, amelynek a cluster kulcsa 3 oszlopból áll. (Vigyázat!!! Több tábla is lehet rajta)

```
SELECT owner, cluster_name FROM dba_clu_columns
GROUP BY owner, cluster_name HAVING COUNT(DISTINCT clu_column_name) = 3;
```

HASH CLUSTER: Hány olyan hash cluster van az adatbázisban, amely nem az oracle alapértelmezés szerinti hash függvényén alapul?

SELECT COUNT(*) FROM
(SELECT owner, cluster_name, hash_expression FROM dba_cluster_hash_expressions)

Hozzunk létre egy hash clustert és rajta két táblát, majd szúrjunk be a táblákba sorokat úgy, hogy a két táblának 2-2 sora ugyanabba a blokkba kerüljön. Ellenőrizzük is egy lekérdezéssel, hogy a 4 sor valóban ugyanabban a blokkban van-e. (A ROWID lekérdezésével) TIPP: A sorok elhelyezését befolyásolni tudjuk a HASH IS megadásával.

Ismétlő kérdések

(ISMETLO KERDESEK.DOC-BÓL RÉSZLETEK)

Az alábbi objektumok közül melyik hány adatszegmenssel rendelkezhet?

- package: 0
- nem partícionált tábla 1 vagy több
- szekvencia, index-szervezett tábla 1 vagy 2
- nézet 0
- trigger 0
- megszorítás (constraint) 0

Mit jelent a 'B' betű a B-fa nevében?

Balanced = Kiegyensúlyozott

Mi a különbség a következő 3 adatszótár tartalmában? Melyik miket tartalmaz? (DBA_TABLES, ALL_TABLES, USER_TABLES)

Egyre kevesebbet találunk, ezekhez egyre kevesebb jogunk van.

8.
Adatbázisok tervezése, megvalósítása, menedzselése

Ajánlott feladatok plusz pontért: **határidő: nov. 25.**
http://people.inf.elte.hu/nikovits/ABTERV/ajanlott_feladat.txt

Zárhelyi megoldások
(Csak azok amelyek nehezebbek, és érdekesebbek voltak)

A NIKOVITS.ELADASOK2 tábla melyik particójában vannak azok a sorok, amelyek számla-száma 104,105,106.

```
select o.objrct_name, o.subobject_name, szla_szam
from eladasok2 e, dba_objects o
where dbms_rowid.rowid_object(e.rowid = o.data_object and szla_szam in (104,105,106)
```

Lekérdezések végrehajtási terve

Hozza létre a PLAN_TABLE táblát, amibe a lekérdezések végrehajtási tervét tehetjük

```
create table PLAN_TABLE (
    statement_id      varchar2(30),
    plan_id           number,
    timestamp         date,
    remarks           varchar2(4000),
    operation         varchar2(30),
    options           varchar2(255),
    object_node       varchar2(128),
    object_owner      varchar2(30),
    object_name       varchar2(30),
    object_alias      varchar2(65),
    object_instance   numeric,
    object_type       varchar2(30),
    optimizer         varchar2(255),
    search_columns    number,
    id                numeric,
    parent_id         numeric,
    depth             numeric,
    position          numeric,
    cost              numeric,
    cardinality       numeric,
    bytes             numeric,
    other_tag         varchar2(255),
    partition_start   varchar2(255),
    partition_stop    varchar2(255),
    partition_id      numeric,
    other             long,
    distribution      varchar2(30),
    cpu_cost          numeric,
    io_cost           numeric,
    temp_space        numeric,
    access_predicates varchar2(4000),
    filter_predicates varchar2(4000),
    projection        varchar2(4000),
    time              numeric,
```

```
    qblock_name      varchar2(30),
    other_xml        clob
);
```

NIKOVITS felhasználó tulajdonában vannak a következő táblák:

CIKK(ckod, cnev, szin, suly)
SZALLITO(szkod, sznev, statusz, telephely)
PROJEKT(pkod, pnev, helyszin)
SZALLIT(szkod, ckod, pkod, mennyiseg, datum)

A táblákhoz indexek is vannak létrehozva, ezek tulajdonságait a katalógusból nézhetitek meg, ha szükségetek van rá. Adjuk meg a következő lekérdezéseket és a hozzájuk tartozó végrehajtási tervek fa struktúráját. Minden esetben lehet hinteket használni. Adjuk meg a piros cikkekre vonatkozó szállítások összmennyiségét.

```
select sum(s.mennyiseg)
from nikovits.cikk c, nikovits.szallit s
where c.ckod = s.ckod and c.szin = 'piros'

EXPLAIN PLAN SET statement_id='gy7f1'
FOR
select sum(s.mennyiseg)
from nikovits.cikk c, nikovits.szallit s
where c.ckod = s.ckod and c.szin = 'piros';

SELECT LPAD(' ', 2*(level-1))||operation||' '||options||' '||object_name terv,
    access_predicates||' -- '||filter_predicates "feltetel (access--filter)"
FROM plan_table
START WITH id = 0 AND statement_id = 'gy7f1'
CONNECT BY PRIOR id = parent_id AND statement_id = 'gy7f1'
ORDER SIBLINGS BY position;

select plan_table_output from table(dbms_xplan.display('plan_table','gy7f1','basic'));
```

```
---------------------------------------
| Id  | Operation           | Name    |
---------------------------------------
|   0 | SELECT STATEMENT    |         |
|   1 |  SORT AGGREGATE     |         |
|   2 |   HASH JOIN         |         |
|   3 |    TABLE ACCESS FULL| CIKK    |
|   4 |    TABLE ACCESS FULL| SZALLIT |
---------------------------------------
```

a) Adjuk meg úgy a lekérdezést, hogy egyik táblára se használjon indexet az oracle.

```
EXPLAIN PLAN SET statement_id='gy7f2'
FOR
select /*+ NO_INDEX(c) NO_INDEX(s) */ sum(s.mennyiseg)
from nikovits.cikk c, nikovits.szallit s
where c.ckod = s.ckod and c.szin = 'piros';
```

```
---------------------------------------
| Id  | Operation           | Name    |
---------------------------------------
|   0 | SELECT STATEMENT    |         |
|   1 |  SORT AGGREGATE     |         |
|   2 |   HASH JOIN         |         |
|   3 |    TABLE ACCESS FULL| CIKK    |
|   4 |    TABLE ACCESS FULL| SZALLIT |
---------------------------------------
```

b) Adjuk meg úgy a lekérdezést, hogy csak az egyik táblára használjon indexet az oracle.

```
EXPLAIN PLAN SET statement_id='gy7f3'
FOR
select /*+ NO_INDEX(c) INDEX(s) */ sum(s.mennyiseg)
from nikovits.cikk c, nikovits.szallit s
where c.ckod = s.ckod and c.szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','gy7f3','basic'));
```

```
-------------------------------------
| Id  | Operation           | Name    |
-------------------------------------
|   0 | SELECT STATEMENT    |         |
|   1 |  SORT AGGREGATE     |         |
|   2 |   HASH JOIN         |         |
|   3 |    TABLE ACCESS FULL| CIKK    |
|   4 |    TABLE ACCESS FULL| SZALLIT |
-------------------------------------
```

c) Adjuk meg úgy a lekérdezést, hogy mindkét táblára használjon indexet az oracle.

```
EXPLAIN PLAN SET statement_id='gy7f4'
FOR
select /*+ INDEX(c) INDEX(s) */ sum(s.mennyiseg)
from nikovits.cikk c, nikovits.szallit s
where c.ckod = s.ckod and c.szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','gy7f4','serial'));
```

```
-------------------------------------------------
| Id  | Operation                     | Name     |
-------------------------------------------------
|   0 | SELECT STATEMENT              |          |
|   1 |  SORT AGGREGATE               |          |
|   2 |   TABLE ACCESS BY INDEX ROWID | SZALLIT  |
|   3 |    NESTED LOOPS               |          |
|   4 |     TABLE ACCESS BY INDEX ROWID| CIKK    |
|   5 |      INDEX RANGE SCAN         | C_SZIN   |
|   6 |     INDEX RANGE SCAN          | SZT_CKOD |
-------------------------------------------------
```

d) Adjuk meg úgy a lekérdezést, hogy a két táblát SORT-MERGE módszerrel kapcsolja össze.

e) Adjuk meg úgy a lekérdezést, hogy a két táblát NESTED-LOOPS módszerrel kapcsolja össze.

f) Adjuk meg úgy a lekérdezést, hogy a két táblát NESTED-LOOPS módszerrel kapcsolja össze, és ne használjon indexet.

9.
Adatbázisok tervezése, megvalósítása, menedzselése

Elméleti alapozás (hisztogramok)

8 millió sor, 100 sor/blokk és a feltétel „O BETWEEN K1 AND K2” és 200 bejegyzés/blokk. Ha FULL olvasást végzünk akkor 80e blokkolvasás, míg ha INDEX olvasást (ez egy B-fa), akkor 5 blokk ha csak egy sort keresünk. Az INDEX esetén a fa 4 magas (1, 20, 40e, 8M). Viszont ha a sorok 5%-át keresem (40e sor) akkor már az INDEX nem hatékony. Hogy az Oracle tudja, hogy mikor-mit érdemes végezni hisztogramot tárol. (Szélesség alapú hisztogram). Ennek minden része ugyanolyan széles. Ha ennek egy része magas, akkor ott FULL-t, ahol alacsony ott INDEX-et fog használni. Van egy magasság alapú hisztogram is, ez pedig változó szélességű, ezek alapján az Oracle tud dönteni arról mit érdemes tennie.

Lekérdezések a NIKOVITS.CIKK, SZALLITO, PROJEKT, SZALLIT táblák alapján

Adjuk meg azon szállítások összmennyiségét, ahol ckod=1 és szkod=2.

```
select * from szallit where szkod = 2 and ckod = 1;
```

- **Adjuk meg úgy a lekérdezést, hogy ne használjon indexet.**
  ```
  select /*+ NO_INDEX(sz) */ * from szallit sz where sz.szkod = 2 and sz.ckod = 1;
  ```

- **A végrehajtási tervben két indexet használjon, és képezze a sorazonosítók metszetét (AND-EQUAL).**
  ```
  select /*+ INDEX(sz) AND_EQUAL(sz) */ * from szallit sz where sz.szkod = 2 and sz.ckod = 1;
  ```

Hozzunk létre 10 intervallumos magasság alapú hisztogrammot az alábbi eloszlású adatokra vonatkozóan, vagyis adjuk meg az egyes intervallumok végpontjait. (Az előfordulások száma zárójelben szerepel.)

1-100 (2), 101-300 (1), 301-400 (4), 401 (200)

```
1        40       80       120 …    … 280   320      360      401
     80       80       60       40      100     160      360
```

Hozzunk létre 10 intervallumos szélesség (gyakoriság) alapú hisztogrammot is a fenti adatokra.

```
1   50  120   200     300 325    400  401
```

Hozzatok létre egy saját példányt a nikovits.szallit táblából, majd adjatok meg egy olyan lekérdezést, amelyik egy általatok választott napra vonatkozóan a szállítások összmennyiségét adja meg. Statisztikák létrehozásával illetve törlésével érjétek el (hint használata nélkül), hogy az Oracle egyszer használjon indexet, máskor pedig (hisztogram létrehozása vagy törlése után) ne használjon.

```
/* Tábla és index létrehozása */
create table szallit as select * from nikovits.szallit
create index idxszallit on szallit(datum);

/* Alap lekérdezés, hogy kitudjunk választani alacsony darabszámú, és magas sorszámú sorokat */
select datum, count(datum) from szallit group by datum order by count(datum) asc;

/* Statisztika törlése */
call dbms_stats.delete_column_stats( 'faboaai', 'szallit', 'datum' );

/* Statisztika létrehozása */
begin
  dbms_stats.gather_table_stats( 'faboaai', 'szallit', method_opt => 'FOR COLUMNS datum SIZE 130' );
end;
```

```
/

/* Statisztika kiirása */
select table_name, column_name, num_distinct, low_value, high_value, num_nulls, density,
num_buckets, histogram
from user_tab_columns where table_name = 'SZALLIT';

/* Listázási parancsok */
select sum(mennyiseg) from szallit where datum = to_date( '2003-12-01', 'YYYY-mm-dd' );
select sum(mennyiseg) from szallit where datum = to_date( '2003-05-01', 'YYYY-mm-dd' );
```

Lekérdezések az Oracle demo táblák alapján (lásd sample_tables.txt)

Adjuk meg azoknak a vevőknek a nevét (SH.CUSTOMERS), akik nőneműek (cust_gender = 'F') és szinglik (cust_marital_status = 'single'), vagy 1917 és 1920 között születtek.

```
select *
from sh.customers c
where ( c.cust_gender = 'F' and c.cust_marital_status = 'single' ) or
  c.cust_year_of_birth in (1917, 1918, 1919, 1920);
```

- **Vegyük rá az Oracle-t, hogy a meglévő bitmap indexek alapján érje el a tábla sorait.**

```
select /*+ INDEX_COMBINE(c) */ *
from sh.customers c
where ( c.cust_gender = 'F' and c.cust_marital_status = 'single' ) or
  c.cust_year_of_birth in (1917, 1918, 1919, 1920);
```

- **Vegyük rá, hogy ne használja ezeket az indexeket.**

```
select /*+ NO_INDEX(c) */ *
from sh.customers c
where ( c.cust_gender = 'F' and c.cust_marital_status = 'single' ) or
  c.cust_year_of_birth in (1917, 1918, 1919, 1920);
```

TERV (OPERATION + OPTIONS + OBJECT_NAME)

Adjunk meg egy olyan lekérdezést az sh tábláira (hintekkel együtt ha szükséges), aminek az alábbi lesz a végrehajtási terve:

```
SELECT STATEMENT +  +
 SORT + ORDER BY +
  TABLE ACCESS + BY INDEX ROWID + CUSTOMERS
   BITMAP CONVERSION + TO ROWIDS +
    BITMAP AND +  +
     BITMAP INDEX + SINGLE VALUE + CUSTOMERS_MARITAL_BIX
     BITMAP OR +  +
      BITMAP INDEX + SINGLE VALUE + CUSTOMERS_YOB_BIX
      BITMAP INDEX + SINGLE VALUE + CUSTOMERS_YOB_BIX
      BITMAP INDEX + SINGLE VALUE + CUSTOMERS_YOB_BIX
```

```
select *
from sh.customers c
where c.cust_marital_status = 'married' and c.cust_year_of_birth in ( 1951,1953,1955)
order by c.cust_id;
```

```
EXPLAIN PLAN SET statement_id='gy8f1'
FOR
 select *
 from sh.customers c
 where c.cust_marital_status = 'married' and c.cust_year_of_birth in ( 1951,1953,1955)
 order by c.cust_id;

select plan_table_output from table(dbms_xplan.display('plan_table','gy8f1','basic'));
```

```
-------------------------------------------------------------------
| Id  | Operation                      | Name                     |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                          |
|   1 |  SORT ORDER BY                 |                          |
|   2 |   TABLE ACCESS BY INDEX ROWID  | CUSTOMERS                |
|   3 |    BITMAP CONVERSION TO ROWIDS |                          |
|   4 |     BITMAP AND                 |                          |
|   5 |      BITMAP INDEX SINGLE VALUE | CUSTOMERS_MARITAL_BIX    |
|   6 |      BITMAP OR                 |                          |
|   7 |       BITMAP INDEX SINGLE VALUE| CUSTOMERS_YOB_BIX        |
|   8 |       BITMAP INDEX SINGLE VALUE| CUSTOMERS_YOB_BIX        |
|   9 |       BITMAP INDEX SINGLE VALUE| CUSTOMERS_YOB_BIX        |
-------------------------------------------------------------------
```

Objektum relációs lekérdezések a példa adatbázis tábláiból
(hr.countries, hr.employees, oe.customers, oe.jobs)

A táblák összekapcsolásának felderítésében segítenek a definiált idegen kulcsok.

Adjuk meg, hogy hány különböző országból vannak vevőink (oe.customers).

Adjuk meg, hogy melyik országból hány vevőnk van. (oe.customers, hr.countries)
Az olyan ország is szerepeljen (ha van), amelyik nincs benne a hr.countries táblában.
(Ország_azon, Országnév, VevőkSzáma)

Adjuk meg azoknak a dolgozóknak a nevét és foglalkozását, akik vevőkkel foglalkoznak,
valamint azt, hogy hány különböző országbeli vevővel foglalkoznak.
(Név, Foglalkozás, OrszágokSzáma)

Adjuk meg a Burt Spielberg nevű vevőnek a telefonszámait (külön sorokban).

Adjuk meg azoknak a német vevőknek a nevét, akiknek legalább 2 telefonszámuk van.

Adjuk meg azoknak az amerikai városoknak a neveit, amelyek az északi szélesség 42. és 43. foka
valamint a nyugati hosszúság 82. és 85. foka között helyezkednek el.
lásd -> oe.customers.cust_geo_location tárolja a vevő városának földi elhelyezkedését.
Az x és y attribútumokban van a földrajzi hosszúság (-180 - 180, a negatív számok Greenwich-től nyugatra)
illetve szélesség (-90 - 90, a negatív számok az egyenlítőtől délre).

Hozzunk létre egy T táblát az OE.ORDERS, OE.CUSTOMERS táblák adatai alapján a következőképpen.
A T táblában legyen benne a vevo neve (first name + last name), címe (cust_address), telefonszámai

(VARRAY oszlopként), valamint az általa feladott rendelések adatai (beágyazott táblaként). A rendelések információi közül elég az azonosítót, dátumot és összértéket (order_id, order_date, order_total) tárolni.

Töltsük is fel adatokkal a T táblát a fenti két táblából. Az összes vevő összes rendelésével kerüljön bele a táblába.

Majd adjuk meg a T táblát lekérdezve azon vevők nevét és összes rendeléseik összértékét, akik címének városa Detriot vagy Indianapilis.

Ellenőrzésképpen adjuk meg ugyanezt a lekérdezést az eredeti két táblára megfogalmazva.