

Adatbázisok 2 **gyakorlat - Brányi**

Dokumentáció/HELP: <http://medusa.inf.elte.hu/oradoc11/index.htm>

Adatbázis objektumok (DBA_OBJECTS)

```
select * from user_objects;  
create table puff (A integer);  
select * from dba_objects where owner='BRANYI' and object_type='VIEW';
```

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

```
select * from dba_objects where object_name='DBA_TABLES' and object_type='VIEW';  
select * from dba_objects where object_name='DUAL' and object_type='TABLE';  
--korrekt megoldása:  
select owner from dba_objects where object_name='DBA_TABLES' and object_type='VIEW';
```

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

-- (Az iménti két lekérdezés megmagyarázza, hogy miért tudjuk elérni őket.)

```
select * from dba_objects where object_name='DBA_TABLES' and object_type='SYNONYM';  
select * from dba_objects where object_name='DUAL' and object_type='SYNONYM';
```

```
select 1+1 from dual; --dual táblában csak 1 sor van.  
select 1+1 eredmény from dual; --a mező neve itt már eredmény :D  
select 1+1 from sila.emp; --minden oszlopban megcsinálja.
```

--Milyen típusú objektumai vannak az orauser nevű felhasználónak az adatbázisban?

```
select object_type from dba_objects where owner='ORAUSER';  
select distinct object_type from dba_objects where owner='ORAUSER'; --ne legyenek többször  
az azonosak.
```

--Melyek ezek a típusok? Hány különböző típusú objektum van nyilvántartva az adatbázisban?

```
select distinct object_type from dba_objects;  
select count(distinct object_type) from dba_objects; --különbözőek: 45  
select distinct count(object_type) from dba_objects; --összes: 72583
```

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

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

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

```
select owner from dba_objects where object_type='VIEW'  
intersect
```

```
select owner from dba_objects where object_type='TRIGGER';
```

--Kik azok a felhasználók, akiknek van nézete, de nincs triggerre?

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

minus

```
select owner from dba_objects where object_type='TRIGGER';
```

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

```
select owner from dba_objects where object_type='TABLE' group by owner having count(*)>40  
intersect
```

```
select owner from dba_objects where object_type='INDEX' group by owner having count(*)<=37;
```

--feltételezzük, hogy van indexe, de mi van akkor, ha egyáltalán nincs?

```
select * from user_tab_columns;
```

```
select * from dba_tab_columns;
```

Táblák oszlopai (DBA_TAB_COLUMNS)

--Hány oszlopa van a nikovits.emp táblának?

```
select * from dba_tab_columns where owner='NIKOVITS' AND table_name='EMP';
```

```
select max(column_id) from dba_tab_columns where owner='NIKOVITS' AND  
table_name='EMP';
```

--Milyen típusú a nikovits.emp tábla 6. oszlopa?

```
select data_type from dba_tab_columns where owner='NIKOVITS' AND table_name='EMP' AND  
column_id=6;
```

```
select data_type || '('||data_precision||','||data_scale||')' from dba_tab_columns where  
owner='NIKOVITS' AND table_name='EMP' AND column_id=6;
```

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

```
select distinct owner, table_name from dba_tab_columns where column_name LIKE 'Z%';
```

-- (LIKE '_' - egy karakter '%' tetszőleges mennyiségű char)

-- select user from dual where 'alma%' LIKE '%!%' escape '!';

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

```
select table_name from dba_tab_columns where data_type='DATE' group by table_name having  
count(*)>=8;
```

--////////////////

```
select * from sh.times;
desc SH.times;
select count(*) from SH.sales;
--//////////
```

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

```
-- count(*)=1879
select table_name from dba_tab_columns where data_type='VARCHAR2' AND column_id=1
intersect
select table_name from dba_tab_columns where data_type='VARCHAR2' AND column_id=4;
```

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.

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

```
select file_name, tablespace_name from dba_data_files
union
select file_name, tablespace_name from dba_temp_files;
```

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

```
select tablespace_name, count(*), sum(bytes) from dba_temp_files group by tablespace_name
order by 3 desc;
```

--Mekkora a blokkok merete a USERS táblatéren?

```
select * from dba_data_files;
select bytes/blocks from dba_data_files where tablespace_name='USERS';
select block_size from dba_tablespaces where tablespace_name='USERS';
select * 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 vegyük figyelembe.)

```
select * from dba_segments;
```

```
select segment_name, bytes from dba_segments where segment_type='TABLE' order by bytes desc;
select * from (select segment_name, bytes from dba_segments where segment_type='TABLE' order by bytes desc) where rownum=1;
select segment_name, bytes from dba_segments where bytes=(select max(bytes) from dba_segments where segment_type='TABLE') and segment_type='TABLE';
```

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

```
-----
SELECT * FROM DBA_SYNONYMS;
SELECT * FROM DBA_VIEWS;
```

```
create table sz from (select * from SILA.szeret);
```

```
create view almaevok as select nev from sz where gy='alma';
select * from almaevok;
```

```
SELECT * FROM DBA_VIEWS where owner='BRANYI';
```

~~~~~

Tomx:

```
SELECT * FROM szinonima1;
```

--majd derítsük ki, hogy kinek melyik tábláját kérdeztük le.

```
select * from dba_objects where object_name='SZINONIMA1';
select * from dba_synonyms where owner='PUBLIC' and synonym_name='SZINONIMA1';
```

```
select * from dba_objects where owner='ORAUSER' and object_name='NEZET1';
select * from dba_views where owner='ORAUSER' and view_name='NEZET1';
```

```
select * from dba_objects where owner='HR' and object_name='EMPLOYEES';
select * from HR.employees;
```

~~~~~

ABlinux:

--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 értékét is módosítsuk az új értékre.

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 TOMX adatbázis
NIKOVITS.FOLYOK ORADB adatbázis

--- Mely folyók érintik Csehországot?

```
select * from nikovits.folyok where orszagok like '%cz%';  
create database link t1link connect to s8o1xf identified by s8o1xf using  
'tomx.inf.elte.hu:1521/ora11g';  
select * from nikovits.folyok where orszagok like '%'||(select tld from  
nikovits.vilag_orzagai@t1link where nev like '%Csehország%')||'%';
```

--- Mely országokon folyik keresztül a Nílus? Az országokat a megfelelő sorrendben adjuk meg (HF) → meo később

ROWID adattípus formátuma és jelentése (lásd még DBMS_ROWID package)

Az egyes blokkokban hány sor van?

```
select * from nikovits.cikk;  
select rownum, nikovits.cikk.* from nikovits.cikk;  
select rowid, nikovits.cikk.* from nikovits.cikk;
```

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 blocks from dba_segments where owner='NIKOVITS' and segment_name='CIKK' and  
segment_type='TABLE'; -- eredmény: 8  
select blocks from dba_tables where owner='NIKOVITS' and table_name='CIKK'; -- mo: 7  
select count (distinct substr(rowid,1,15)) from nikovits.cikk; -- eredmény: 4
```

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 substr(rowid,1,15), count(*) from NIKOVITS.cikk  
group by substr(rowid,1,15) order by 1;
```

```
select count(distinct substr(rowid,1,15)) from NIKOVITS.cikk group by substr(rowid,1,15) order by 1;
```

Hozunk létre egy táblát az EXAMPLE táblatéren, amelynek szerkezete azonos a nikovits.cikk tábláéval és pontosan 128 KB helyet foglal az adatbázisban. Foglajunk 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 masolat
tablespace EXAMPLE
storage(initial 128K)
as select * from NIKOVITS.cikk where 0=1; --table MASOLAT created.
alter table masolat allocate extent (size 128k); --table MASOLAT altered.
```

```
DECLARE
  x number;
BEGIN
  FOR i IN 1 .. 1000 LOOP
    insert into masolat select * from nikovits.cikk where i=ckod;
    select count(distinct substr(rowid,1,15)) into x from masolat;
    EXIT WHEN x>1;
  END LOOP;
END;
/
```

```
select substr(rowid,1,15), count(*) from masolat group by substr(rowid,1,15) order by 1; --309
```

```
select * from dba_tab_columns where owner='S8O1XF' and table_name='MASOLAT';
```

Próbáljuk ki az előzőt ismét, de most a PCTFREE értéket állítsuk 40-re. Mindkét esetben ellenőrizzük is, hogy a sorok tényleg két blokkban vannak, és a másodikban csak egyetlen sor van.

```
create table masolat
tablespace EXAMPLE
storage(initial 128K)
PCTFREE 40
as select * from NIKOVITS.cikk where 0=1;
select * from masolat;
alter table masolat allocate extent (size 128k);
```

```
select count(distinct substr(rowid,1,15)) from NIKOVITS.cikk group by substr(rowid,1,15) order by 1;
```

```

DECLARE
  x number;
BEGIN
  FOR i IN 1 .. 1000 LOOP
    insert into masolat select * from nikovits.cikk where i=ckod;
    select count(distinct substr(rowid,1,15)) into x from masolat;
    EXIT WHEN x>1;
  END LOOP;
END;
/

```

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 * from SH.SALES;
SELECT ROWID from sh.sales where prod_id=13 and cust_id=35834 and
time_id='98-JAN.-10'; --AAAWISAAEAAACISAAS
SELECT ROWID from sh.sales where prod_id=13 and cust_id=35834 and
time_id=to_date('1998.01.10','yyyy.mm.dd'); --AAAWISAAEAAACISAAS

```

papiros feladat: Kiterjeszthető hasító index (vagy másképp kiterjeszthető tördelőtáblázat) ----- A kosártömb mérete mindig pontosan 2^i . Tegyük fel, hogy egy blokkba 2 rekord fér el, j értéke (a blokkok jobb oldalán) azt jelzi, hogy hány bit használatos a blokkhoz tartozás eldöntésére. Szűrjük be az alábbi hasító értékkel rendelkező sorokat egymás után, és minden újabb blokk létrehozása után rajzoljuk újra a kosártömböt és a blokkokat: 0011, 0110, 1011, 0111, 1110, 1111, 0100

Lekérdezések (DBA_INDEXES, DBA_IND_COLUMNS, DBA_IND_EXPRESSIONS)

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.

```

select * from dba_indexes;
select * from user_indexes;

```

```

create table sz as select * from branyi.sz;
create index nevsor on SZ(N);

```

```
create index nevgyumolcs on SZ(N,GY);
create index nevharmadik on SZ(substr(N,3,1));
create index nevesharmadik on SZ(N,substr(N,3,1));
create index nevsorcsokkeno on SZ(N desc);
```

```
select * from dba_ind_columns where index_owner='S8O1XF';
select * from user_ind_columns; -- itt 5 van
```

```
select * from dba_ind_expressions where index_owner='S8O1XF';
select * from user_ind_expressions; --itt 3 van (substr fv)
```

--harmadik gomb: milyen index-szel jut a megoldáshoz

```
select * from sz where n='Kanga';
select * from sz where gy='alma';
```

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

```
select * from dba_ind_columns where descend='DESC';
select distinct table_name 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 * from dba_ind_columns where column_position=9;
```

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

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

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

```
select index_owner, index_name from dba_ind_columns group by index_owner, index_name
having count(*)=2
```

intersect

```
select index_owner, index_name from DBA_IND_EXPRESSIONS;
```

--- Mely országokon folyik keresztül a Nílus? Az országokat a megfelelő sorrendben adjuk meg (HF)

```
select nev,instr((select orszagok from nikovits.folyok where nev='Nilus'),tld)
      from NIKOVITS.vilag_orzagai@tlink
where instr((select orszagok from nikovits.folyok where nev='Nilus'),tld)>0 order by 2;
```


@tlink ez a kapcsolat amit csináltál a tomix-ra

Lexy:

```
-----
select nev,instr((select orszagok from nikovits.folyok where nev='Nilus'),tld)
from NIKOVITS.vilag_orszagai@mytomxlink
where instr((select orszagok from nikovits.folyok where nev='Nilus'),tld) > 0 order by 2;
--@mytomxlink ez a kapcsolat amit csináltál a tomix-ra
--create database link mytomxlink connect to fgytvf identified by fgytvf using
'tomx.inf.elte.hu:1521/ora11g';
```

```
select instr ('0101110000010101110','00',1,4) from dual;
```

```
SELECT * FROM NIKOVITS.vilag_orszagai@mytomxlink;
SELECT * FROM NIKOVITS.folyok;
```

```
select nev,instr((select orszagok from nikovits.folyok where nev='Nilus'),tld)
from NIKOVITS.vilag_orszagai@mytomxlink
where instr((select orszagok from nikovits.folyok where nev='Nilus'),tld) > 0 order by 2;
```

--PARTITION fajták:

--http://people.inf.elte.hu/nikovits/AB2/cr_part_table.txt

--http://people.inf.elte.hu/branyi/ora/gyak2/t06/abterv_feladatok6.txt

Particionálás (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 * 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 * 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 * from dba_part_key_columns where owner='SH' and name='COSTS' and
```

object_type='TABLE';

--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 = 'ELADASOK3' or table_owner = 'SH' and
table_name = 'COSTS')
and partition_position = 2;
```

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

--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;
```

--??? (table_segments, dba_part_tables, dba_tab_partitions)

```
select * from dba_objects where owner='NIKOVITS' AND object_name='ELADASOK'; --4 db van
select * from dba_objects where owner='NIKOVITS' AND object_name='ELADASOK4'; --13 db
van → alpartíciók...???
```

ZH-n: szünet után :L

papíros 45 perc, gépes 45 perc (minuszos nem, rendszertáblák iger

Hajas leírás összeszedve [ITTEN](#)

Lexy Jegyzete a ZH-ra:

Példák [prefix]re:

USER (az adott user tulajdonában lévő objektumok) pl. USER_TABLES

ALL (amihez joga van az adott usernek) pl. ALL_TABLES

DBA (az adatbázis összes objektuma) pl. DBA_TABLES

A legfontosabb adatszótár, ahonnan érdemes kiindulni a kereséskor:

DBA_OBJECTS (ALL_OBJECTS)

[prefix]_TAB_COLUMNS : infó az egyes táblák oszlopairól.

- owner: tulajdonos neve
- table_name : tábla neve
- column_name : oszlop neve
- data_type : Milyen típusú elemek vannak az oszlopban?
- column_id : Az oszlop ID-ja (hányadik oszlop ez az adott táblában?)
- data_precision : Number típusúnál a pontosság (milyen hosszú?)
- data_scale: Number típusúnál a Skálázás (Hány tizedes jegy?)

DBA_TABLES : táblákkal kapcsolatos információk.

DBA_DATA_FILES : Milyen adatfájlok vannak

DBA_TEMP_FILES : hasonló mint a Data files csak ebben Temporális fájlokkal kapcsolatos információk

DBA_TABLESPACES : táblaterekre van osztva az adatbázis, erről való infók

DBA_SEGMENTS : Adatbázis objektumok fizikai tárolásával kapcsolatos információk.

DBA_EXTENTS : Segments extensekből áll

DBA_FREE_SPACE: szabad helyek, egy adott fájlban melyik blokkban mennyi szabad hely van.

DESCRIBE szamok; -- egy tábla oszlopainak és azok típusának kiírása

```
select * from masolat;
```

```
select      table_name,column_name,column_id,data_precision,data_scale,data_      from
user_tab_columns;
```

--Milyen típusú a nikovits.emp tábla 6. oszlopa?

```
select data_type from dba_tab_columns where owner='NIKOVITS' AND table_name='EMP' AND
column_id=6;
```

```
select * from dba_tab_columns where owner='NIKOVITS' AND table_name='EMP' AND
column_id=6;
```

```
select data_type || '('||data_precision||','||data_scale||')'
from dba_tab_columns
where owner='NIKOVITS' AND table_name='EMP' AND column_id=6;
```

```
select * from DBA_DATA_FILES;
```

```
select * from dba_segments where owner = 'SH';
```

--melyik azok a táblák amelyek fel vannak particionálva?

--szegmens a tábla fizikai megjelenése

--particionálnak minden részének van egy szegmense
select * from dba_part_tables where owner = 'SH';

--tábla fizikai elhelyezkedése, lehet index tábla is!
--ha particionált a tábla
--valamelyik tábla melyik fájlban helyezkedik el?
--> az ebben található File ID
--> relative_fno: általában megegyezik a File ID-val

--kérdezzük le valamelyik SALES tábla mérete az SH ownertől
--count* :
select distinct segment_type from dba_segments;
--table: egy particiókba álló táblát jelenti ezért a count(*) = 1

--particionált táblák mérete, és a particiók száma
select segment_name,sum(bytes) size_partiton,count(*) name_partitons
from dba_segments
where (segment_type = 'TABLE PARTITION' or segment_type = 'TABLE SUBPARTITION')
group by segment_name;

DESCRIBE számok;

Lexy

https://docs.google.com/document/d/1HBCOkwqmbbZ_00JrQIGmye73L36_WTFboZS6OQHRVqk/edit?usp=sharing



9. gyakorlat

Algebrai Optimalizálás (csak Vizsgán!)

<http://people.inf.elte.hu/kiss/13ab2oszl/13ab2oszl.htm>

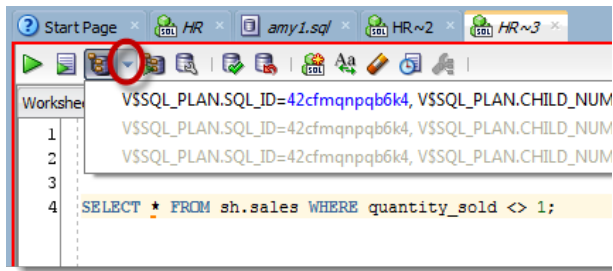
Lekérdezések optimalizálásának feladata: [algebrai opt.ppt](#) (91-101 oldal)

http://people.inf.elte.hu/branyi/ora/gyak2/t05/abterv_feladatok5_uj.html

<http://people.inf.elte.hu/nikovits/AB2/>

- [UW_egymenetes_algoritmusok.doc](#)

- [UW_hash_algoritmusok.doc](#)



<http://people.inf.elte.hu/nikovits/AB2/utlxplan.sql>

<http://people.inf.elte.hu/nikovits/AB2/tervek4.txt>

<http://people.inf.elte.hu/nikovits/AB2/tervek3.txt>

ZH: megkapjuk az alsó rajzot és ki kell találni mi volt a feladat

explain plan for

select * from Sila.emp;

explain plan for

select ename,sal from Sila.emp;

explain plan set

statement_id='nesze' for

select ename,sal from Sila.emp where ename='KING';

select * from plan_table;

ahány sor stimmel annyi pontos a feladat...???

Számít hogy melyik oracle-n !

(Ki lesz kötve pl hogy Tomx-en)

+++++

10. gyakorlat

<http://people.inf.elte.hu/nikovits/AB2EA/naplo.ppt>

http://people.inf.elte.hu/nikovits/AB2EA/naplo_osszefoglalas.txt

UNDO naplózás

<T,Start> <start T> <T, Begin>

<T, vált, érték>

<T,commit> --ügyfél megkapta amit akart (feladat végrehajtva, végleges állapot a lemezen)

<T, abort>

<T,end>

REDO naplózás (helyrehozó)

<T,commit> (az ügyfél megmondta mit akar, majd egyszer megkapja amit akar.- kirúgjuk) a lemezreírás csak utána

<T,end> (az ügyfél megkapta amit akar, lemezreírás)

<END CKPT> minden ckpt előtti commitolt adatot endelt

UNDO-REDO naplózás

<T, vált, régi, új>

fentről lefelé kiírás

visszaírás alulról felfelé

end ckpt csak akkor ha minden függőben lévő adatokat letároljuk

start előttiakat eltároljuk

select * from emp natural join dept; → végrerhajtási terv

select /*+use_nl(emp dept) */ * from emp natural join dept; → végrerhajtási terv

select /*+use_merge(emp dept) */ * from emp natural join dept; → végrerhajtási terv

select /*+use_hash(emp dept) */ * from emp natural join dept; → végrerhajtási terv

ablinux: van index -- merge jointtal megy

tomix: nincs index -- hash jointtal megy

ablinux-on van egy primary key a dept táblában (tomix-en nincs) és gyártott egy indexet.

HF:

<http://people.inf.elte.hu/nikovits/AB2/tervek1.txt>

<http://people.inf.elte.hu/nikovits/AB2/tervek2.txt>

<http://people.inf.elte.hu/nikovits/AB2/tervek3.txt>

<http://people.inf.elte.hu/nikovits/AB2/tervek4.txt>



11. gyakorlat

<http://people.inf.elte.hu/nikovits/AB2EA/konkurencia.ppt>

http://people.inf.elte.hu/sila/edu13sep/ab2_gy12.html

konkurencia -

Milyen átrendezéseket végezhetünk el a tranzakciók között?

rövidítések :

r = read

w = write

r1(x)

1 - tranzakció száma

x - adat

r1(A) w1(A) r2(A) w2(A) r1(B) w1(B) r2(B) w2(B)

Lehet:

Két egymás melletti olvasást szabad cserélni (nem ugyanabban a tranzakcióban!)

Írás-olvasás különböző tranzakcióban, ha nem ugyan azt ír, amit a másik olvas

Írás-írás különb tranzakció, ha különböző adatot írnak szabad

Nem lehet:

ugyanabban a tranzakcióban cserélni

ugyanazt az adatot nem lehet egyszerre írni-írni vagy írni-olvasni

/* A win 8-ban a Jeg -beírásakor eltűnik a jegyzetömb, mert 'g' betű lett lenyomva, és nem 'gy' (y-nál visszajön) - pl a putty-nál meg 'tty' betűt kéne leírni, hogy végig látszódjon a keresésben */

megelőzési gráf :

páronként össze kell nézni a tranzakciók műveleteit, és megnézni ezek alapján, hogy a tranzakciókat milyen sorrendben kell elvégezni

(DAG - topologikus rendezés - while(van csúcs) {mindig a nulla befokú csúcsokat kiírjuk, majd kitöröljük a belőlük kimenő éleket & a csúcsokat})

Lockolások:

li(a)

Lockoljuk az i. tranzakcióban az A adatot

ui(a)

Unlockoljuk ...

2PL

Első megoldás : Lőjük le a legrégebbit (hogy commitoljon)

Miért nem lehet Kanga Robi? - Mert le van lockolva.

xli(x) - exclusive lock - csak a mienk az erőforrás, más nem nyúlhat hozzá

ZH maiból pl : mi lesz a tranzakciók megelőzési gráfja? Sorba rakható-e?

megelőzési gráf

hint (mit használjon a lekérésnél ... (nested loop, hash, sort merge))

[illegible]

12. gyakorlat

http://people.inf.elte.hu/nikovits/AB2/ab2_feladat6.txt

```
create table szallit as select * from nikovits.szallit;
```



```
select /*+ no_index(cikk) no_index(szallit) */ sum(mennyiseg) from cikk natural join szallit where
szin='piros';
```

```
create index szall on szallit(ckod);
```

```
// szallit szall indexét használja
```

```
select /*+ no_index(cikk) index(szallit szall) */ sum(mennyiseg) from cikk natural join szallit
where szin='piros';
```

```
ne legyen space a /*+ között
```

```
// merge
```

```
select /*+ use_merge(cikk szallit) no_index(cikk) index(szallit szall) */ sum(mennyiseg) from cikk
natural join szallit where szin='piros';
```

```
// nested loop
```

```
select /*+ use_nl(cikk szallit) no_index(cikk) index(szallit szall) */ sum(mennyiseg) from cikk
natural join szallit where szin='piros';
```

```
select /*+ use_nl(cikk szallit) */ * from cikk,szallit;
```

```
select /*+ use_merge(cikk szallit) */ * from cikk,szallit;
```

```
select /*+ use_hash(cikk szallit) */ * from cikk,szallit;
```

```
// ez utolsó helyett nestedloopot csinál. Miért nem lehet rávenni? - Mert értelmetlen a kosarazás
(hisz mindenki mindenkivel...)
```

de 'natural join' - nál mindegyiket be lehet állítani

(<http://people.inf.elte.hu/nikovits/AB2/expl.sql>)

```
explain plan for (select *...);
```

```
select plan_table_output from
```

```
table(dbms_xplan.display('plan_table',null,'all'));
```

Minta adatok (http://people.inf.elte.hu/nikovits/AB2/sample_tables.txt)

lehet, hogy ezekből (eredetikből) lesz a zh

Papíros :

Naplózás 3, költség számítás, megelőzési gráf

egy kis UNDO naplózás példa:

(http://people.inf.elte.hu/sila/edu13sep/ab2_gy10.html#Tk.8.2_undo_naplozas)

Valamivel probléma van, szóljunk neki (most itt órán)

/* minél előbb, amint megtudjk a zh-jegyet, kell nyilatkozni, hogy jó-e, mert csak ha gyakjegy a Neptunba van benn akkor mehetünk vizsgázni. */

gépes kb a 3. így fog kinézni (amit vissza kell fejteni)

```
SELECT szállito.szkod, projekt.pkod, telephely
```

```
FROM szállito, szállit, projekt, cikk
```

```
WHERE szállit.szkod=szállito.szkod
```

```
AND szállit.pkod=projekt.pkod
```

```
GROUP BY szállito.szkod, projekt.pkod, telephely
```

```
HAVING SUM(mennyiség) > 500;
```

/* Brányi nagyon dölöngél, és unja a semmittevést, majdhogynem bealszik */

Össze kéne ülni tanulni ?

ZH:

3 végrehajtási terv

GYakorlásnak:

Géptermi:

http://people.inf.elte.hu/nikovits/AB2/ab2_feladat7.txt

PApiros:

http://people.inf.elte.hu/nikovits/AB2/ab2_feladat8.txt

http://people.inf.elte.hu/nikovits/AB2/ab2_feladat9.txt