7-8. gyak

7. gyak

```
/*****************************/

create table PLAN_TABLE (...); --expl.txt

DROP TABLE PLAN_TABLE;

EXPLAIN PLAN SET statement_id='ut1' -- ut1 -> az utasításnak egyedi nevet adunk
FOR
SELECT avg(fizetes) FROM nikovits.dolgozo;

SELECT * FROM PLAN_TABLE;

SELECT LPAD(' ', 2*(level-1))||operation||' + '||options||' + '||object_name terv
FROM plan_table
START WITH id = 0 AND statement_id = 'ut1' -- az utasítás neve szerepel itt
CONNECT BY PRIOR id = parent_id AND statement_id = 'ut1' -- meg itt
ORDER SIBLINGS BY position;


SELECT SUBSTR(LPAD(' ', 2*(LEVEL-1))||operation||' + '||options||' + '||object_name,
    1, 50) terv,
cost, cardinality, bytes, io_cost, cpu_cost
FROM plan_table
START WITH ID = 0 AND STATEMENT_ID = 'ut1' -- az utasítás neve szerepel itt
CONNECT BY PRIOR id = parent_id AND statement_id = 'ut1' -- meg itt
ORDER SIBLINGS BY position;

select plan_table_output from table(dbms_xplan.display('plan_table','ut1','all'));
select plan_table_output from table(dbms_xplan.display());

CREATE TABLE dolgozo AS SELECT * FROM nikovits.dolgozo;
CREATE TABLE osztaly AS SELECT * FROM nikovits.osztaly;
CREATE TABLE Fiz_kategoria AS SELECT * FROM nikovits.Fiz_kategoria;

EXPLAIN PLAN SET statement_id='ut2' -- ut1 -> az utasításnak egyedi nevet adunk
FOR
SELECT DISTINCT onev
FROM dolgozo NATURAL JOIN osztaly
WHERE fizetes > (
SELECT also
FROM FIZ_KATEGORIA
WHERE kategoria = 1
) AND fizetes <= (
SELECT felso
FROM FIZ_KATEGORIA
WHERE kategoria = 1
);

CREATE INDEX ind_onev
ON osztaly(onev);

select plan_table_output from table(dbms_xplan.display('plan_table','ut1','all'));
```

```
select plan_table_output from table(dbms_xplan.display('plan_table','ut2','all'));

---=== 1. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy egyik táblára se használjon indexet az oracle.
EXPLAIN PLAN SET statement_id = 'cikk_orig'
FOR
SELECT sum(mennyiseg)
FROM nikovits.szallit NATURAL JOIN nikovits.cikk
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','cikk_orig','all')
    );

---=== 2. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy csak az egyik táblára használjon indexet az
    oracle.
EXPLAIN PLAN SET statement_id = 'cikk_index_one_table'
FOR
SELECT /*+ index(c) */ sum(mennyiseg)
FROM nikovits.szallit NATURAL JOIN nikovits.cikk c
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','
    cikk_index_one_table','all'));

---=== 3. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy mindkét táblára használjon indexet az oracle.
EXPLAIN PLAN SET statement_id = 'cikk_index_two_tables'
FOR
SELECT /*+ index(c) index(sz) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','
    cikk_index_two_tables','all'));

---=== 4. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy a két táblát SORT-MERGE módszerrel kapcsolja ö
    ssze.

EXPLAIN PLAN SET statement_id = 'cikk_sort_merge'
FOR
SELECT /*+ USE_MERGE(sz c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','cikk_sort_merge',
    'all'));

---=== 5. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy a két táblát NESTED-LOOPS módszerrel kapcsolja ö
    ssze.

EXPLAIN PLAN SET statement_id = 'cikk_nested_loops'
FOR
SELECT /*+ USE_NL(sz c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
```

```
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','cikk_nested_loops
    ','all'));

---=== 6. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy a két táblát HASH-JOIN módszerrel kapcsolja össze
    .

EXPLAIN PLAN SET statement_id = 'cikk_hash_join'
FOR
SELECT /*+ USE_HASH(sz c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','cikk_hash_join','
    all'));

---=== 7. feladat ===---
-- Adjuk meg úgy a lekérdezést, hogy a két táblát NESTED-LOOPS módszerrel kapcsolja ö
    ssze,
-- és ne használjon indexet.

EXPLAIN PLAN SET statement_id = 'cikk_nested_loops_no_index'
FOR
SELECT /*+ no_index(sz) no_index(c) USE_NL(sz c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE szin = 'piros';

select plan_table_output from table(dbms_xplan.display('plan_table','
    cikk_nested_loops_no_index','all'));
```

8. gyak

```
---=== 1. feladat ===---
-- Adjuk meg azon szállítások összmennyiségét, ahol ckod=2 és szkod=2.
EXPLAIN PLAN SET statement_id = 'cikk2_orig'
FOR
SELECT sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 2 AND szkod = 2;

select plan_table_output from table(dbms_xplan.display('plan_table','cikk2_orig','all'
    ));

-- Adjuk meg úgy a lekérdezést, hogy ne használjon indexet.
EXPLAIN PLAN SET statement_id = 'cikk2_no_index'
FOR
SELECT /*+ no_index(sz) no_index(c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 2 AND szkod = 2;

select plan_table_output from table(dbms_xplan.display('plan_table','cikk2_no_index','
    all'));
```

```
-- A végrehajtási tervben két indexet használjon, és képezze a sorazonosítók metszetét
     (AND-EQUAL).
-- nem jó!
EXPLAIN PLAN SET statement_id = 'cikk2_and_equal'
FOR
SELECT /*+ AND_EQUAL(sz) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 2 AND szkod = 2;

select plan_table_output from table(dbms_xplan.display('plan_table','cikk2_and_equal',
    'all'));

---=== 2. feladat ===---
-- Adjuk meg a Pecs-i telephely? szállítók által szállított piros cikkek összmennyiség
    ét.

EXPLAIN PLAN SET statement_id = 'cikk3_orig'
FOR
SELECT sum(mennyiseg)
FROM nikovits.szallit NATURAL JOIN nikovits.cikk NATURAL JOIN nikovits.szallito
WHERE szin = 'piros' AND telephely = 'Pecs';

select plan_table_output from table(dbms_xplan.display('plan_table','cikk3_orig','all'
    ));

-- Adjuk meg úgy a lekérdezést, hogy a szallit táblát el?ször a cikk táblával join-
    olja az oracle.
EXPLAIN PLAN SET statement_id = 'cikk3_order_change'
FOR
SELECT /*+ leading(sz c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c NATURAL JOIN nikovits.szallito
    szo
WHERE szin = 'piros' AND telephely = 'Pecs';

select plan_table_output from table(dbms_xplan.display('plan_table','
    cikk3_order_change','all'));

-- Adjuk meg úgy a lekérdezést, hogy a szallit táblát el?ször a szallito táblával join
    -olja az oracle.
EXPLAIN PLAN SET statement_id = 'cikk3_order_change2'
FOR
SELECT /*+ leading(sz szo) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c NATURAL JOIN nikovits.szallito
    szo
WHERE szin = 'piros' AND telephely = 'Pecs';

select plan_table_output from table(dbms_xplan.display('plan_table','
    cikk3_order_change2','all'));

---=== 3. feladat ===---
-- Adjuk meg azon szállítások összmennyiségét, ahol ckod=1 vagy szkod=2.
EXPLAIN PLAN SET statement_id = 'cikk4_orig'
FOR
SELECT sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 1 OR szkod = 2;
```

```
select plan_table_output from table(dbms_xplan.display('plan_table','cikk4_orig','all'
    ));

--Adjuk meg úgy a lekérdezést, hogy ne használjon indexet.
EXPLAIN PLAN SET statement_id = 'cikk4_no_index'
FOR
SELECT /*+ no_index(sz) no_index(c) */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 1 OR szkod = 2;

select plan_table_output from table(dbms_xplan.display('plan_table','cikk4_no_index','
    all'));

-- A végrehajtási tervben két indexet használjon, és képezze a kapott sorok unióját (
    CONCATENATION).
EXPLAIN PLAN SET statement_id = 'cikk4_concat'
FOR
SELECT /*+ index(sz) index(c) USE_CONCAT */ sum(mennyiseg)
FROM nikovits.szallit sz NATURAL JOIN nikovits.cikk c
WHERE ckod = 1 OR szkod = 2;

select plan_table_output from table(dbms_xplan.display('plan_table','cikk4_concat','
    all'));
```