

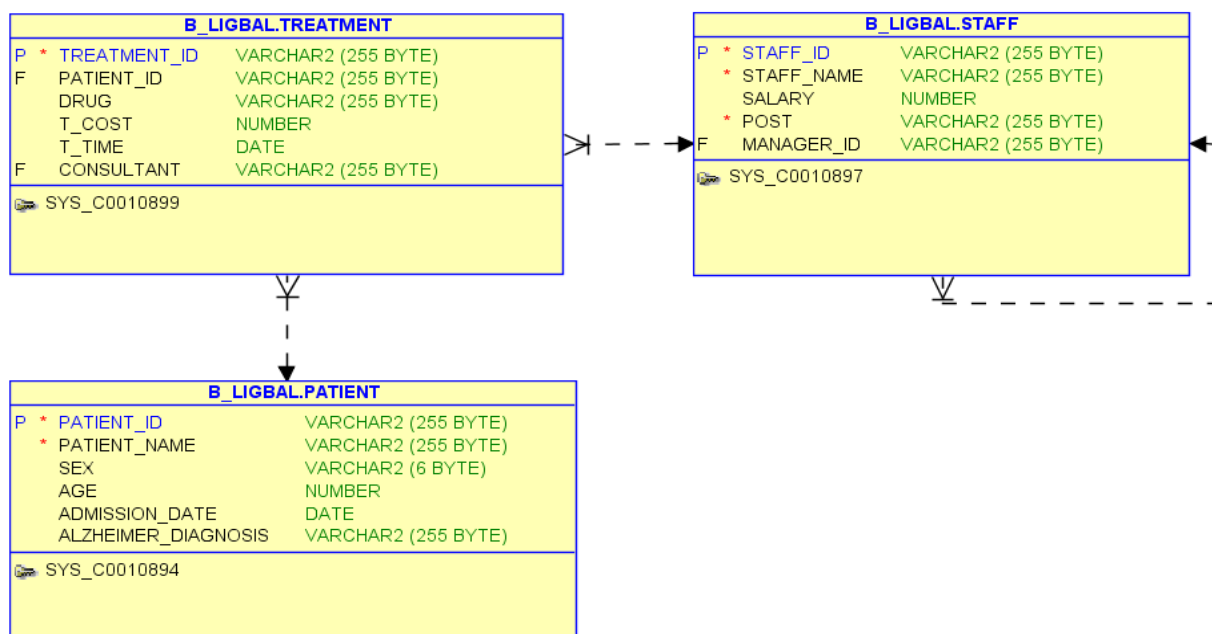
### 3. gyakorlat

Töltsd le a tárgy wiki oldaláról a `gyak02_mb_dbs_2016.sql` fájlt, majd ezt a szkript fájlt futtasd le az sql developerben!

Link:

[https://wiki.itk.ppke.hu/twiki/pub/PPKE/AdatbazisKezelesMB2018/gyak02\\_mb\\_dbs\\_2018.s  
ql](https://wiki.itk.ppke.hu/twiki/pub/PPKE/AdatbazisKezelesMB2018/gyak02_mb_dbs_2018.sql)

Relációs séma:



## I. Join

### 1. JOIN

Az összetett kérdéseinkre gyakran nem tudunk válaszolni 1 tábla segítségével, mert a különböző entitásokhoz tartozó információkat praktikus okokból külön táblákban tároljuk. A táblák összekapcsolását JOIN segítségével tehetjük meg, valamilyen olyan attribútum alapján, ami mindkét táblában megtalálható (ez jellemzően egy elsődleges kulcs és egy külső kulcs páros).

Tehát ha az érdekel minket, hogy egy adott kezelést milyen orvos végzett el (mi az orvos neve, egyéb adatai), akkor szükségünk van a *treatment* és a *staff* táblára is. Itt a közös attribútum a kezelést végző orvos id-ja. A *treatment.consultant* a külső kulcs, és a *staff.staff\_id* az elsődleges kulcs. Ekkor a join így néz ki:

```
SELECT treatment.treatment_id,  
       staff.staff_name  
FROM   treatment  
INNER JOIN staff  
ON treatment.consultant = staff.staff_id;
```

1. **Feladat.** Melyik kezelést ki végezte és mennyibe került? Az eredményt rendezd kezelés ára alapján csökkenő sorrendbe!
2. **Feladat** Módosítsd úgy az előző lekérdezést, hogy azokat is kiírja, akik nem végeztek kezelést!
3. **Feladat.** Melyik beteget ki kezelte és mikor?
4. **Feladat.** Kinek ki a közvetlen főnöke?

## II. Egyszerű halmaz műveletek

### 2. Distinct

Ha arra vagyunk kíváncsiak, hogy egy oszlopban (vagy oszlopokban) milyen egymástól eltérő értékek vannak, akkor a SELECT DISTINCT konstrukciót használhatjuk. → megszünteti a duplikációkat.

Szintaxis:

```
SELECT DISTINCT column_name, column_name  
FROM table_name;
```

Pl. A különböző nevű dolgozók listája:

```
SELECT distinct name  
FROM dolgozo
```

5. **Feladat.** Az eddig kezelések során milyen gyógyszereket használtak?
6. **Feladat.** Listázd ki a betegek és az orvosok ID-ját, duplikációk nélkül!

## III. Union, intersect, minus

Szintén gyakori, hogy valamilyen eredményeknek az uniójára, metszetére vagy különbségére vagyunk kíváncsiak.

```
SELECT column_name(s) FROM table1  
UNION  
SELECT column_name(s) FROM table2;
```

7. **Feladat.** Ki az, aki nem végzett egyetlen kezelést sem?
8. **Feladat.** Ki az, aki főnöke valakinek, de nem ő az igazgató?
9. **Feladat.** Listázd ki az összes kórházban lévő ember nevét, és azt, hogy beteg-e, vagy alkalmazott!

## IV. Egymásba ágyazott lekérdezések, VIEW

### VIEW

Egy lekérdezés (SELECT) eredményére tekinthetünk úgy, mint egy „táblára”. Azaz magát a query-t eltárolhatjuk egy virtuális táblaként.

```
CREATE OR REPLACE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

Ekkor már a view-ból is le lehet kérdezni, úgy mintha az egy tábla volna:

```
SELECT column_name(s)
FROM view_name
WHERE condition
```

pl.

```
CREATE OR REPLACE VIEW KivegezteAkezelest AS
SELECT
t0.treatment_id,
t1.STAFF_NAME,
t0.t_cost
FROM treatment t0
INNER JOIN staff t1
ON t1.STAFF_ID = t0.CONSULTANT;
```

- 10. Feladat.** Hozz létre egy view-t a következő lekérdezésekhez: melyik beteget ki kezelte és mikor?
- 11. Feladat.** A nézet használatának segítségével válaszold meg, hogy mely betegeket kezelte Dr. Green 2004 októberében.

## V. Lekérdezések a from, where, stb klózban

Mivel a lekérdezés eredménye egy „tábla”, ezért ezeket felhasználhatjuk más lekérdezésekben is.

- 12. Feladat.** Melyik dolgozónak mennyivel több vagy kevesebb a fizetése, mint az átlagos fizetés?
- 13. Feladat.** Melyek azok az orvosok, aki végeztek már legalább egy kezelést?