

2. gyakorlat

Ismétlés

Fogalmak: SQL, DDL, DML,

Bemelegítő feladat:

- Hozd létre a következő táblát,
- állítsd be a kulcsokat,
- töltsd fel az alábbi értékekkel
- majd válaszd ki azon gyümölcsök azonosítóját, és mennyiségét, ahol a mennyiség nagyobb, mint 10 tonna.

fruits

fruit_id	f_name	color	best_before	amount_in_tons
F1500	apple	red	24-10-2004	21
F9700	peach	peach	02-08-2010	5
F9500	apple	green	04-09-2010	32

Új anyag

I. SELECT

Ha csillagot írunk (**attribnevek helyett**), akkor mindent felsorol.

SELECT attribnevek

FROM tablanevek

[WHERE feltételek];

Részletesebben:

SELECT *oszloplista* (mit szeretnénk látni a kimeneten - projekció)

FROM *táblalista* (miből szeretnénk látni, eddig a kötelező rész)

WHERE *logikai kifejezés* (mivel szűrünk - szelekció)

ORDER BY *logikai rendezés* (csökkenő-növekvő sorba rendez)

Példa:

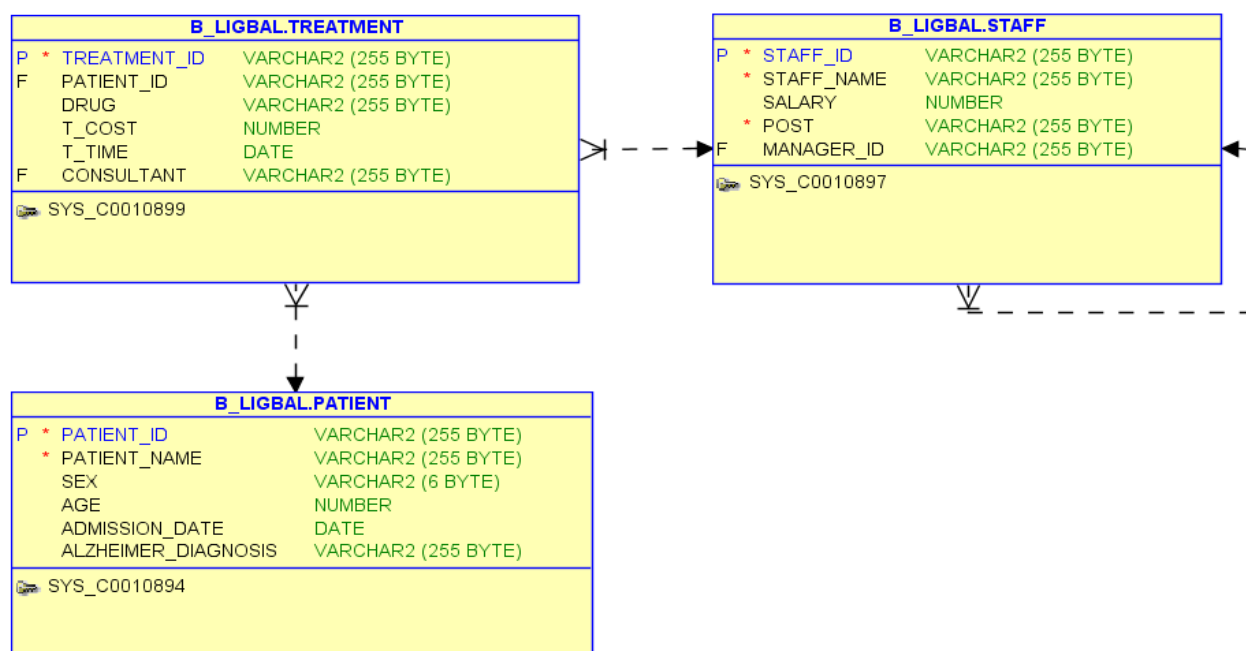
SELECT * FROM dolgozo;

Töltsd le a tárgy wiki oldaláról a `gyak02_mb_dbs_2018.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:



0. Feladat. Listázd ki milyen táblákat érsz el! (user_tables)

- 1. Feladat.** Listázd ki a betegeket az aktuális életkorukkal („age” mező: életkor az első felvételnél, Aktuális dátum: SYSDATE)
- 2. Feladat.** Listázd ki azokat a pacienseket koruk szerint csökkenő, sorrendbe rendezve (azonos kor esetén névsorban), akik 2005.01.01. után kerültek felvételre!

II. UPDATE, DELETE

UPDATE

Ha valamilyen recordot frissítenünk kell, pl. megváltozott a beteg neve, akkor az UPDATE parancsot lehet használni:

UPDATE table_name

SET column1=value1,column2=value2,...

WHERE some_column=some_value;

Példa:

```
UPDATE dolgozo  
SET name = 'Géza'  
WHERE name = 'Károly'
```

3. **Feladat.** P9500 azonosítójú beteg diagnózisa megváltozott moderate-ről severe-re. Ennek megfelelően frissítsd az adatbázist!
4. **Feladat.** Opcionális: Frissítsd az életkort úgy, hogy az aktuális életkort mutassa, ne pedig a felvételkorit! (Aktuális dátum: SYSDATE)

DELETE

Szintén gyakori, hogy egy rekordot, vagy rekordokat törölni kell az adatbázisból.

```
DELETE FROM table_name  
WHERE some_column=some_value;  
Példa: Az összes Károly nevű dolgozó törlése:
```

```
DELETE FROM dolgozo  
WHERE name = 'Károly'
```

5. **Feladat.** Töröld az összes nőnemű beteget!
6. **Feladat.** A Henry Kip nevű beteget áthelyezték, töröld az adatbázisból!
7. **Feladat.** Töröld az össze rekordot a táblából! (Where klóz elhagyható)

III. WHERE klóz

1. Pattern matching

Akkor használjuk, ha valamilyen karaktermintázatot akarunk megtalálni. Például azt kérdezzük, hogy mely dolgozókat hívják Károlynak. Ezt fejezhetjük ki a LIKE kulcsszóval.

'%' – tetszőleges és bármilyen hosszúságú stringgel való egyezés (Figyelem, nem a pl. kereséseknél vagy reguláris kifejezésekben szokásos '*')

'_' – egyetlen karakterben való egyezés (Figyelem, nem a szokásos '?')!

Pl.:

```
SELECT *  
FROM dolgozo  
WHERE name LIKE '%Károly%'
```

Ha arra vagyunk kíváncsiak, hogy melyek azok a nevek, amik 'A' betűvel kezdődnek és a kettővel rákövetkező karakter is 'A':

```
SELECT *  
FROM dolgozo  
WHERE name LIKE 'A_A%'
```

Ha az összes rekordot törölted korábban, akkor a wikin lévő sql script alapján visszatöltheted az adatokat a táblába.

- 8. Feladat.** Mik az 'ar' vagy 'Ar' karaktereket tartalmazó betegek adatai?
9. Feladat. Mi az azonosítója és a kora a ' Henry' stringgel kezdődő betegeknek?

2. Null értékek kezelése

SQL-ben, ha csak máshogy nem rendelkezünk, lehetnek 'NULL' értékek. Ekkor azonban az összehasonlító operátorokat (<>,=,<) és a függvényeket óvatosan kell használni. Erre megoldás az IS NOT NULL és az IS NULL használata.

```
SELECT *  
FROM patient  
WHERE name IS NULL
```

- 10. Feladat.** Válasszátok ki azokat a sorokat a treatment táblából, amelyek cost attribútuma nem 'üres'!
11. Feladat. Válasszátok ki az összes rekordot, de úgy, hogy a Cost mezőben hiányzó értékeket 0-val helyettesítitek! (Használd az nvl() függvényt)
12. Feladat. Mit ad vissza a következő lekérdezés?

```
SELECT *  
FROM treatment  
WHERE t_cost < 60 OR t_cost >= 60;
```

IV. Egyszerű függvények, join

1. Egyszerű függvények

Alapvetően két típusba csoportosíthatjuk a függvényeket: aggregáló vagy skaláris. Az aggregáló függvények jellemzően 1 db értékkel térnek vissza, amit az adott oszlopban lévő összes adatból számolnak ki. Az aggregáló függvényekre példa:

- a) avg() – Az átlaggal tér vissza
- b) count() – A sorok számával tér vissza
- c) first() – Az első értékkel tér vissza
- d) last() – Az utolsó értékkel tér vissza
- e) max() – A legnagyobb értékkel tér vissza
- f) min() – A legkisebb értékkel tér vissza
- g) sum() – A sorok összegével tér vissza

A skaláris függvények ezzel szemben jellemzően egy értéket rendelnek hozzá valamilyen inputhoz. Ilyen pl:

- a) `to_date()` – Dátum típusra konvertál egy stringet
- b) `to_number()` – Számra konvertál egy stringet
- c) `lower()` – Stringből csupa kisbetűs stringet állít elő
- d) `upper()` – Stringből csupa nagybetűs stringet állít elő
- e) `nvl()` – Nullokat cseréli valamilyen megadott értékre
- f) `round()` – Számot kerekít

13. Feladat. Írd ki a gyógyszerek nevét kisbetűvel!

14. Feladat. Mennyi az átlag fizetés?

15. Feladat. Egy hónapban a kórház mennyi pénzt költ a dolgozók fizetésére?

16. Feladat. Mikor volt a legkorábbi és a legkésőbbi kezelés?

17. Feladat. Hány kezelés volt összesen?

18. Feladat. Hányszor adtak gyógyszert összesen?

19. Feladat. Opcionális: mi a betegek vezeték - és keresztnéve? Milyen függvények kellenek ehhez? (használd a google-t)