# 8. gyakorlat (3. 31.)

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

Laptopom továbbra sincs, géptermi gépen jegyzetelek, amennyire lehet. UPDATE: Szörnyű…

ToDo:

A kezdeti jelszó csak egy rövid ideig érvényes, és a későbbiekben is bizonyos idő után lejár. A lejárati idő lekérdezhető a következő utasítással:  
SELECT username, account\_status, expiry\_date FROM DBA\_USERS;

Kérek mindenkit, hogy a jelszavát változtassa meg az alábbi SQL utasítással!  
ALTER USER <usernév> IDENTIFIED BY <új\_jelszó>;

## Dátumfüggvények

Hónap hozzáadása, stb. lehetséges.

SELECT CURRENT\_DATE FROM DUAL;

CURRENT\_TIMESTAMP, SYSDATE, SYSDATE\_TIMESTAMP

Mit ad vissza: év, hó, nap, óra, perc, másodperc, időzóna

Csak Oracle tudja:

SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;

Lehet kerekíteni (pl. évekre), levágni, stb. (ROUND)

## Konverziós függvények

TOCHAR TONUMBER stb.

TODATE Dátum sokszor gondot okoz, bevitelnél is.

TOCHAR zero padding, decimális pont, elválasztás minden harmadiknál. Karakterből is lehet karaktert konvertálni (pl.: nemzetiségi)

## CASE

CASE kif

WHEN kif\_1 THEN kif\_1

…

WHEN kif\_n THEN kif\_n

[ELSE kif]

END

CASE

WHEN logikai kif THEN

…

END

SELECT SIN(SAL) FROM EMP;

<->

SELECT CASE JOB WHEN 'PRESIDENT' THEN 'Elnök' WHEN 'CLERK' THEN 'Titkár' ELSE 'Egyéb' END FROM EMP;

SELECT CASE SAL WHEN SAL<1000 THEN 'Kevés' WHEN SAL<3000 THEN 'Közepes' ELSE 'Sok' END FROM EMP;

## Dekódolás

DECODE(s, kif\_1, kife\_1, kif\_2, kife\_2, … kif\_k, kife\_k, … egyéb)

SELECT DECODE(JOB, 'PRESIDENT ', 'ELNÖK ', 'CLERK ', 'TITKÁR ', 'EGYÉB ') FROM EMP;

## NULL kezelés

SELECT ENAME, SAL, COMM, SAL+COMM FROM EMP;

Az összes SAL+COMM átvált NULL-ra, ahol nincs komisszió (jutalék)

Függvény nélkül:

SELECT ENAME, SAL, COMM, SAL+COMM FROM EMP WHERE COMM IS NOT NULL UNION SELECT ENAME, SAL, COMM, SAL FROM EMP WHERE COMM IS NULL;

Függvénnyel:

SELECT ENAME, SAL, COMM, SAL+NVL(COMM,0) FROM EMP;

Ne a NULL-t, hanem a nullát add össze. x+NULL helyett x+0 a kifejezés.

Az ANSI szabványban nem NVL, hanem COALESCE a függvény, többet is tud, mint az Oracle. (NVL csak NULL-t cserélni, COALESCE minden értéket.)

Saját nevünk, nem fogjuk használni:

SELECT USER FROM DUAL;

## Adattípusok

Nincs BOOLEAN, programozás közben fog megjelenni. Helyette használjunk INTEGER-t.

Rengeteg PostgreSQL-specific adattípus van, IP cím, MAC address, stb. Még PATH is van, ami egy egész útvonal, törésekkel (GPS-hez nagyon hasznos)

SQLite-ben csak 5 típus van: INTEGER TEXT REAL, stb. Nem bonyolítja túl.

Oracle: CHAR, VARCHAR2 (Mindenhol máshol VARCHAR a neve!), NUMBER, DATE, TIMESTAMP

Az összes ANSI kompatibilis szót mind elfogadja, különben nem lenne a szabvánnyal kompatibilis. De nem használ más típust, mint a fenti 4-et. Nagy átverés.

Hiába kérjük, hogy 2 bájtos SMALL\_INT legyen, az adatbázisban 30 bájtos NUMBER lesz belőle. Nem spórolunk semmit.

## Tábla másolása

CREATE TABLE táblanév AS SELECT … annak megfelelően, hogy mit szeretnénk lemásolni.

Helyi másolat a hivatalos EMP tábláról:

CREATE TABLE EMP AS SELECT \* FROM SILA.EMP;

Nem az egészet másolom le:

CREATE TABLE EMP2 AS SELECT ENAME, JOB FROM SILA.EMP;

Ha nem kellenek adatok, csak a struktúra:

CREATE TABLE EMP3 AS SELECT \* FROM SILA.EMP WHERE 1=2;

## Tábla létrehozása

CREATE TABLE táblanév

(mezőnév\_1 típus\_1 [kezdőérték\_1] [[constraint sormegszorítás\_elnevezés\_1] sormegszorítás\_1]

[, mezőnév\_2 típus\_2 [kezdőérték\_2] [sormegszorítás\_2]]

[…]

[, mezőnév\_n típus\_n [kezdőérték\_n] [sormegszorítás\_n]]

[, táblamegszorítás]);

## Megszorítás (sor és tábla)

NULL – Amelyik adattípus (…)

NOT NULL

UNIQUE

UNIQUE ()

PRIMARY KEY

PRIMARY KEY ()

REFERENCES

UNIQUE KEY

CHECK

Példa:

CREATE TABLE adatok

(telefonszám varchar2(20) null,

sorszám number(6) primary key,

becenév char(100) unique,

nem char(1) check(nem='f' or nem='m' or lower(nem)='s'

születés date default sysdate not null,

fizetés number(10,3) null);

(Kis/nagy betűvel is elfogadjuk a semleges nemet.)

Telefonszám miért varchar? Mert külföld hívása 00-val kezdődik, előhívószám +36, stb.) Üresen lehet hagyni.

Sorszámot nem lehet üresen hagyni: not null

De ennél többet akarunk; egyedi legyen: unique

Még többet akarunk: egyedi azonosító

unique lehet null? Igen, mert null sosem egyenlő null-lal. Erre jó a unique not null megszorítás. Vagy a primary key, de azt csak egy helyre lehet betenni.

default sysdate miatt a not null sosem fog kiakadni, mert rendszerdátumot megkapja.

## A megszorításokat el lehet nevezni

telefonszám varchar2(20) constraint tel\_megszorítás null

## Referencia

Főnök nélkül nem vehető fel új alkalmazott:

[…], Manager NUMBER(4) REFERENCES EMP (EMPNO) […]

Először a vezérigazgatót vesszük fel, akinek a főnöke NULL, utána az ő közvetlen beosztottjait (területi igazgatók), majd azok beosztottjait. Sorrend számít.

## Táblamegszorítás

Soronként rossz lenne:

keresztnév UNIQUE

vezetéknév UNIQUE

Így "Kis András" kilövi az összes családtagját (Kis), sőt, az összes Andrást is.

Egész táblára:

UNIQUE (keresztnév vezetéknév)

Így már jó. Kis András csak még egy Kis Andrást nem enged felvenni.

A primary key is működik így. A felsorolt oszlopok együtt határoznak meg egy egyedi sort.

Check több oszlopra: fizetés például nagyobb legyen, mint a jutalék.

Foreign key csak táblára használható:

Foreign key (DEPTNO) references dept (deptno)

## Nézettáblák

CREATE VIEW

Meglévő táblát úgy nézni, hogy bizonyos dolgokat nem szeretnénk látni.

CREATE VIEW nevek as select n from sz;

A "szeret" táblának csak a neveit tartalmazza.

CREATE VIEW almát\_szeret as select n from sz where gy='alma';

Az almát szeretők nevei.

Nézettáblába lehet bevinni adatot?

insert into nevek values('Bolha');

Igen, de az eredeti táblába kerül, nem a nézettáblába. Mivel csak nevet vittem be, a gyümölcs NULL marad.

insert into almát\_szeret values('Kutya');

Be engedi vinni az almát szeretők nézettáblán keresztül, de nem fogja automatikusan szeretni az almát. Sőt:

CREATE VIEW almát\_szeret2 as select \* from sz where gy='alma';

insert into almát\_szeret2 values('Elefánt', 'Banán');

Rossz (nem alma) gyümölccsel is felvehető.

Tehát: Lehet nézettáblába bevinni adatot, de:

• Ha csak olvasási jogunk van az eredeti táblára, akkor nem.

• create view kereset as select ename, sal+nvl(comm,0) from emp;

insert into kereset values('valaki', 10000);

Virtuális oszlopba (sal+comm) nem lehet behelyezni.

Ez már akkor is probléma, ha megszorítjuk egy NOT NULL-lal az eredeti gyümölcs oszlopot.

## Táblatörlés

drop table táblanév

## DESC

Tábláról információt ír ki. (Oszlopok, típusok, NOT NULL megszorítás hol van)

Gyakorlat vége.