Gyakorlati óravázlat 01

Zsolt útmutatója alapján készítette BHG (2017. január 31.)

# Tárgykövetelmények ismertetése, ismerkedés (15-20 perc)

* Bemutatkozás, fontosabb adatok megadása (elérhetőség: email, moodle, ami szimpatikus)
  + Használják a hallgatói fórumot
  + Kérdezzenek!
  + Töltsenek fel fényképet
* Névsorolvasás – nem oda tartozó személyek elküldése könyörtelenül (nincs elég hely)
* Moodle-ben mit hol találnak: előadás kurzus elején sok hasznos információ, linkek, tantárgyi adatlap (kéretik tanulmányozni!), gyakorlati kurzusban mi található
* Gyakorlatok rendjének ismertetése
  + hiányozni nem praktikus, mert úgy nehéz lesz összegyűjteni a tesztpontokat
  + aki késik, annak saját felelőssége felvenni a fonalat
  + Moodle a Te barátod, minden ott van fenn
* Sprint pontok: alkalmanként 5 pont, modulonként összesítve nézzük a %-ot
  + első héten az adott heti előadásból és gyakorlatból, jövő héttől kezdve mindig az előző hétből, illetve egy-egy ismétlő kérdés is előfordulhat. (a tesztkérdések egy része a vizsgán a beugró tesztben visszaköszönhet)
  + **40%** alatt nem lesz vitt pontjuk a vizsgára
  + **80%**-tól jelvény jár
  + a sprintet a saját gyakorlati kurzusukban írják, ott tudják követni a pontszámukat is. (???)

# Tényleges gyakorlati munka (max 60 perc)

1. Volt-e kérdés az előadáson elhangzottakkal kapcsolatosan? – remélhetőleg nem.
2. SQL Developer rövid ismertetése (hogy elmagyarázzuk, miért ezzel fogunk dolgozni, megismerkedni)
   * Egy egyetemi szerverre telepített oktatási célú Oracle12c adatbázissal fogunk dolgozni egész félévben (Oracle a leginkább meghatározó piaci szereplő – 30% körül, az SQL nyelv ugyan standard elvileg, de mégis vannak nyelvjárások: Oracle, IBM DB2 – közel 30%-os penetrációval nagyon hasonlít és még az adattárházaknál gyakori Teradata nyelvjárása is ezekhez hasonlít, Microsoft SQL eléggé különbözik, mySQL-lel van közös nyelvcsaládban – MsSQL világ szinten kb. 10%)
   * A kliens programunk az SQL Developer lesz: Java alapú, IDE környezet (Integrated Development Environment, azaz Integrált Fejlesztői Környezetet).
   * Könnyű telepíteni (portable változat is van), egyszerűbb használni a normál kliens programoknál, mindenhol fut, ahol fut a Java
   * ingyenes. Otthonra is.
   * Most már adatbányász eszköz is van benne; kapcsolható majdnem minden adatbázishoz, amihez van Java alapú konnektor (Ms Accesshez csak a korábbi verziók)
   * Egy nagyon elterjedt kliens program, felhasználói felület még az sql\*plus: teljesen fapados, mindent parancssorosan lehet csak megoldani, cserébe teljesen bug mentes és helyesen kezeli a jogosultságokat minden esetben. Sok cégnél a DBA-k csak ezen keresztül kezelhetik az adatbázist. Többek között ezért is fontos, hogy ugyan megnézünk bizonyos menüket majd a Developerben, de meg kell tanulni az SQL utasításokat, illetve a dokumentáció használatát, hogy akkor is boldoguljunk, ha valamit nem tudunk fejből (programozáskor is sql-t kell írnunk, nem tudunk a programban összekattintgatni). A vizsgán is az sql kódokat kérjük számon.
   * letöltés [innen](http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html): regisztrációhoz / belépéshez:
     1. bce\_hallgato@freemail.hu
     2. GazdInfo.123
     3. ne változtassanak jelszót, mert a többiekkel szúrnak ki.
   * Riport, menedzsment és fejlesztő eszköz – melyik miért fontos (megbeszéljük akkor, amikor már látjuk is): rengeteg féle felhasználó van a végfelhasználókon kívül, ebből megnézzük az üzemeltetést (figyeli a folyamatokat, nem kívánt eseményeket, riportok figyelése) és az adatbázis gazdát (fizikai működésért felel, DBA fül – majd látjuk)
   * Fontos megérteni, hogy nekik otthon csak az sql developert kell telepíteni, azzal otthonról (VPN kapcsolat kell!) ugyanahhoz az adatbázishoz fognak tudni kapcsolódni, mint a géptermekből. Oktatási, tanulási és kutatási célra komplett adatbázis szerver is telepíthető ingyen, aki akar, de a tárgyhoz ez egyáltalán nem szükséges.

Dokumentáció: otn.oracle.com  
(Oracle dokumentáció:

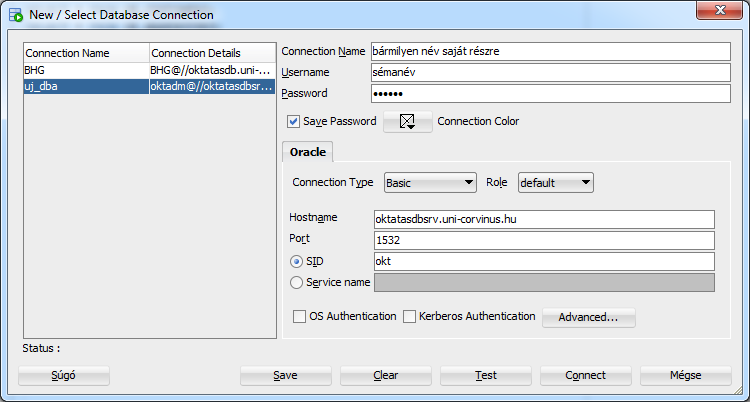
Database SQL Language Quick Reference: <https://docs.oracle.com/database/121/SQLQR/toc.htm>

Database SQL Language Reference: <https://docs.oracle.com/database/121/SQLRF/toc.htm>

SQL Developer dokumnetáció: <http://docs.oracle.com/cd/E55747_01/index.htm> )

– központi kurzusba ki fogom tenni őket. FONTOS, hogy megtanulják ezeket használni, mert fejből képtelenség mindent tudni. Az utolsó heti gyakorlati zh-n, illetve a vizsga gyakorlati részén csak ezt lehet használni (félig open book: saját források nem használhatóak, de az oracle dokumentáció korlátlanul) (A minimum teszthez és az órai tesztekhez semmi nem használható!!!)

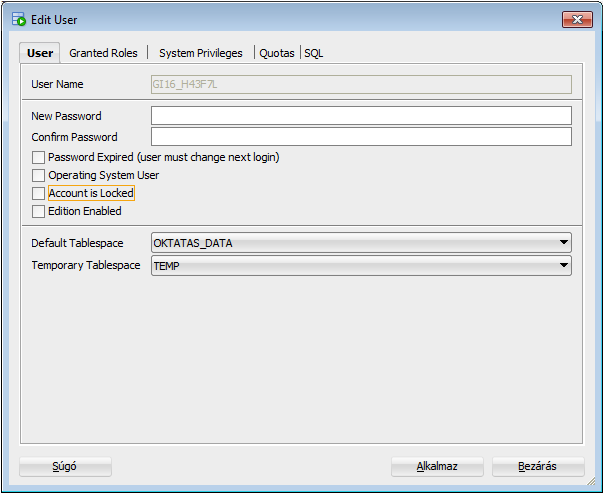
1. SQL Developer indítása („C:\Program Files“ alatt illene legyen)
   * JDBC kapcsolat létrehozása: (jelszó: haho17)



* + 1. Connection name bármi lehet
    2. Username: gi17\_<neptunkód>
    3. Jelszó: haho17
    4. Hostname: egy szabályos domain név: oktatasdbsrv.uni-corvinus.hu
    5. Port: 1532 (ugyanazon a porton is lehet több adatbázis, ezért kell megadni még a következőt is☺
    6. SID: okt (ez a konkrét adatbázis neve, amihez kapcsolódunk)
    7. Service name: nem használjuk, a fenti információkat lehetne egy összetett kifejezéssel itt megadni
    8. Kapcsolat mentésére helyes sorrend: Test → Save → Connect.

beszéljünk egy kicsit a felületről, hogy mi hol van, mi mire való: kapcsolathoz lehet workshhet-et nyitni, illetve a kapcsolat alatt lehet kilistázni az objektumokat. Ez a fejlesztői környezet. A connecitons ablakban lehet menüből, párbeszédablakok segítségével létrehozni, módosítani, törölni dolgokat, a Worksheet-en pedig lehet írni az Sql utasításokat. Amit menüből csinálunk, annak a hátterében is mindig egy sql fut le, átalában ezt meg is tudjuk nézni. Hangsúlyozzuk, hogy az SQL-t fogjuk számon kérni!

1. Jelszóváltás (Connection nevén jobb egér: Reset password...) – nagyon jegyezzék meg, hogy mire változtatták. Mentődik a profiljukban, de ha otthonról akarnak csatlakozni, akkor ahhoz is fog kelleni, illetve ha valamiért nem tölt be a profiljuk, vagy törlődik, akkor is újra be kell állítaniuk (oktatói userünknek van alter user joga, pl. ALTER USER <felhasználó> IDENTIFIED BY <új jelszó>; VAGY: dba 🡪 Security 🡪 Users 🡪 adott user-en jobb egérgomb Edit (itt tudjuk az esetleges lock-ot is megszüntetni)



1. Fejlesztőeszköz része (jövő héttől főleg ezt fogjuk használni)
   * Objektumainkat itt tudjuk gyorsan áttekinteni a Connektions ablakban (még nincs semmink, de az Other users alatt nézzük meg a MINTA objektumait, nem részletesen, csak hogy lássuk, hogy ott már van valami )
   * eredmények futtatása: egyetlen SQL utasításunk ma:   
     select sysdate from dual;   
     nézzük meg a két futtatási módot,   
     
     + Első zöld nyíl (run), gyorsbillentyű: ctrl + Enter vagy F9: mindig csak azt az utasítást futtatja, amiben a kurzor áll (ha több van kijelölve, akkor több fülön fogjuk látni az eredményt), eredmény táblázatos formában a Query Result fülön, kb. Úgy kezelhető mint egy Excel tábla (rendezés, szűrés, oszlopszélesség, oszlopsorrend változtatható), Excelbe exportálható.
     + Második ikon (run script), gyorsbillentyű: F5: a kijelölt utasítások futtatása, vagy az egész worksheet minden utasításának futtatása. Eredmény a Script Output fülön, több utasítás esetén az eredmények illetve visszajelzések egymás alatt.
     + A radír ikon van a Worksheetnél (10.) és a Script outputnál is (mindent töröl), a worksheeten lehet simán is törölni.
     + (9. ikon Aa: lehet a kisbetű-nagybetű rendszeren változtatni)
     + Script output szöveges fájlként menthető. File menüből a worksheetek menthetők szintén szöveges fájlként (sql kiterjesztés). (Az sql developer újranyitásakor a legutóbbi mappa minden .sql fájlját betölti külön worksheetekre) – Érdemes az órai munkájukat mindig lementeni, tanuláshoz hasznos lehet! Illetve az egyes órákhoz én is fogok sql fájlt adni, amit developerből megnyitva helyből lehet futtatni.
     + **(feladat):** Egy minta sql fájl már az első órához: gyak1.sql – nyissuk meg, futtassuk le egyes utasításait. Hajtsuk végre a megjegyzésben leírt módosítást. Mentsük el a módosított worksheet-et.
   * explain plan – tesztelésre ideális, megérteni, hogy mi miért fut úgy, ahogy... lassú-e vagy nem, min érdemes változtatni. (később lehet, hogy látjuk egy gyakorlaton – vagy nem)
   * PL/SQL kódok írása (tárolt eljárások, függvények) (ilyet nem fogunk csinálni, de ezt is itt lehet)
   * Debug funkciók is vannak hozzá...
   * Minden utasítást egy közös kapcsolaton hajt végre alapból az SQL Developer. Fontos lenne elmondani, hogy egy szerveren több adatbázis is lehet (a port mindenképpen különbözik), amelyeknek nevük van. Egy adatbázishoz pedig több fizikai kapcsolat is lehet**. Session = fizikai kapcsolat önálló hálózati portokon keresztül a szerver és a kliens között.**
   * Ha szeretnél, akkor a SQL Worksheet ablakon levő „open unshared worksheet” paranccsal (8. ikon) lehet külön kapcsolatú ablakot nyitni, ami hasznos lehet pl. tranzakciókezeléses dolgok vizsgálatakor – egyelőre ne foglalkozzunk ezen túl vele. (egy sessionhöz is lehet több worksheet-et nyitni, a fülön látszik a különbség, nincs sárga csillag – van sárga csillag)
2. Riport része: itt csak lekérdezések vannak, láthatjuk többek között az éppen aktuális futás paramétereit
   * állapotok megismerése
   * session-höz kötődő paraméterek:
   * **(feladat):** Nézzük meg a Session-öket: Data Dictionary Reports 🡪 Database Administration 🡪 Sessions 🡪 Sessions   
     Látjuk, hogy sok-sok GI17-es usernek van sessionje, látjuk, hogy ezek aktívak, vagy inaktívak, látjuk, hogy a mieinken kívül milyen sessionök vannak még, melyik gépről vannak nyitva, stb. Ha nyitottunk unshared worksheetet, akkor látjuk, hogy két sorral szerepelünk a session listában, majdnem minden egyforma, de az SID tuti különbözik!
   * Monitoring
   * **(feladat):** Nézzük meg a legkimerítőbb SQL futtatások listáját (Reports → Data Dictionary Reports → Database Administration → Top SQL), görgessük el jobbra az ablakot, ott nem csak az utasítás látszik, hanem mindenféle paraméterei is, pl. cpu idő (egy egyszerű egytábláls lekérdezés általában század sőt ezred másodperc nagyságrendben fut le. Itt a lista elején találunk percekig tartó futásokat, illetve a néhány másodpercesek is már nagyon költségesnek számítanak)
   * adatszótár (data dictionary) elemzése   
     Az adatbázis rendszerek fontos jellemzője, hogy **saját magukról is minden adatot az adatbázisban tárolnak**. Az automatikusan tárolt adatok vannak az adatszótárban. Ezekből készített tárolt lekérdezéseket (nézetek) látunk itt a Reports ablakban. Csak az elsőt használjuk belőle. Ott mapparendszerben találhatjuk meg, amit keresünk (meg is írhatjuk helyettük a szükséges lekérdezéseket, illetve meg is nézhetjük, hogy milyen sql-lel jöttek létre) -
3. Mendzsment eszköz: DBA nézet: szerver adminisztráció: szintén látunk lekérdezéseket, emellett itt be is állíthatunk dolgokat menüből (ezeknek is megvan az sql megfelelője, pl. jelszómódosítás)
   * DBA nézet (szerverbeállítások, rendszergazai megoldások)
   * Jogosultságok kezelése

Jogosultsági rendszer felépítése: A jogosultsági rendszer több szempontú: egyik szempont, hogy ki kapja a jogosultságot: lehet egy-egy felhasználónak adni jogosultságokat, vagy felhasználók egy csoportjának (role). Halmazlogikával működik és a szerepeknek is lehet hierarchiája, pl. Az oktato szerep hallgato is. A másik szempont, hogy mire vonatkozik a jogosultság: a jogosultságok objektumokhoz és tevékenységekhez köthetőek. Például a tábla létrehozása is egy jogosultsági elem, de ha fejlesztenénk bármilyen külső funkciót, akkor az is arra a valamire egyedi jogosultságot kapna az Oracle rendszerben és vannak tevékenységhez kapcsolódó jogok (system privileges: pl. A hallgato szerepű felhasználók minden táblát és minden adatszótárat lekérdezhetnek); A harmadik szempont pedig, hogy az adott felhasználó, vagy adott szerepű felhasználó csak a saját maga számára kapta a jogosultságot, vagy tovább is delegálhatja azt: ADMIN option.

* Jogosultsági rendszer működésének rövid bemutatása (View → DBA → connect → Security → Users → <saját felhasználó> → Roles, Sys Privs stb. fülecskék bambulása) illetve (View → DBA → connect → Security → Roles → különböző szerepekhez tartozó jogosultságok áttekintése: nézzük meg az OKTATO, HALLGATO, CONNECT szerepeket. Pl. a hallgato minden jogosultságánál NO szerepel az Admin Option-nál, de az OKTATO pl a HALLGATO szerepet tovább adhatja.
* **(feladat)** – mindenki megkeresi, hogy milyen jogai vannak: fontos, hogy szerepeken keresztül is lehet örökölni jogokat, ott is célszerű megnézni ;).

Fizikai adattárolás felépítése: Szintén több elemű rendszer. Felhasználók csoportjai ún. táblahelyekhez (tablespace) kapcsolódnak (általában a különböző jogosultságú felhasználók adatai fizikailag is elkülönülhetnek), amelyek adatállományokból (data files) állnak. Mindenkinek van munkaterülete és átmeneti tárhelye (TEMP), ahova a lekérdezések részeredményei kerülnek átmeneti jelleggel. Az adatfájlok fizikailag ott vannak a háttértáron, másolhatóak, korlátozott mértékben mozgathatóak. A táblahelyek kihasználtsága az adatállományok kihasználtságának összesítése. Ennél többet erről nem kell tudni.

Az adatbáziskezelő átveszi az operációs rendszertől a fájlkezelési feladatokat (is)

Nem kell tudni és nem kell elmondani, csak Nektek jelzem, hogy az adatállományok adatblokkokból állnak (logikailag egyszerre kezelt = egyszerre a memóriába tehető tárterület – lásd operációs rendszerek ☺), amelyeket ún. extentekbe szervezünk (extent: a háttértáron vagy memóriában összefüggő adatblokkok egy csoportja, amelyet egyszerre kívánunk beolvasni, menteni), ezeket pedig szegmensekké vagy magyarul lapokká vonunk össze. A lapok cél szerint osztályozódnak: vannak index lapok (meglepő, de indexeket tárolunk rajta), adatlapok (itt vannak az adataink), rollback és még sokan mások. A lapok táblához és objektumokhoz kötöttek. Tehát a helyes sorrend: tablespace > adatállomány > szegmens/lap (valójában egy szegmens több adatállományban is lehet) > extent > adatblokk. Azért mondtam el ezeket, mert a feliratokra rákérdezhetnek a hallgatók.

* DBA → Storage → Tablespaces  
  (ha nincs odaírva a mértékegység, akkor általában KB-ban értendő, de ezt dokumentációból le lehet ellenőrizni)
* Fizikai tárolás: DBA → Storage → Datafiles: nézzük meg a használati szinteket
  + **(feladat)**: kattintsunk az egyik fájlra, majd nézzük meg az SQL kifejezést (azt az sql-t fogjuk látni, amivel módosítani lehet a jellemzőit, ha van rá jogosultságunk – nincs ☺)
* Ütemező (DBA → Scheduler) bemutatása. Feladat: önállóan nézzék végig, ha kérdeznek fontosat ☺, válaszoljunk
  + Pontos riport itt: (Reports → Data Dictionary Reports → Schedular → DBMS jobs → DBA jobs
* Adatbázis jelenlegi állapota (DBA → Database Status → Status)
  + verzió lekérdezése
  + milyen operációs rendszeren fut
  + fizikailag hol van a háttértáron
* Használati jellemzők áttekintése (DBA → Database Configuration → View Database Feature Usage) – itt látható, hogy milyen modulok aktívak egy adatbázisban. (pl. data mining nem aktív, automatikus UNDO management igen)

Fontos megjegyezni, (ez előadáson is többször elhangzott tavaly egyébként), hogy a rendszer legkockázatosabb szereplője a DBA, a teljhatalmú úr. Ez a kockázat pl. úgy csökkenthető, ha belőle sem csak egy van, és pl a jogosultságok adására másik DBA-nak van joga, mint egyébként a rendszer karbantartását végző DBA.

# Sprint (5 perc)

(a kérdések az első előadásra illetve az aktuális gyakorlatra vonatkoznak):