

ORACLE andmebaasi praktikum

Sissejuhatus

User Name on S25 (st sügis 2025), millele järgneb kolm perekonnanime esimest tähte ja kolm eesnime esimest tähte. Password on kõigil **AAA** (kolm suurt A-d). **See tuleb kohe alustuseks ära muuta – jätke see uus password siis omale meelde ka :-)** Selleks peate otsima üles SQL-käsu, millega saate oma passwordi ära vahetada - Google (otsing: Oracle change my own password) või allpool olev link aitavad.

Näiteks oleks minu (Raspel Priit) kasutajanimi **S25RASPRI**

Täpitähed on nimedes asendatud vastavate mitte-täpitähtedega (näit: Ä → A, Š → S, Ž → Z jne.)

Oracle SQL-keele (ja ka muu) dokumentatsioon asub veebiaadressil seal on tähestikuline register:
http://docs.oracle.com/cd/E11882_01/server.112/e41084/index.htm

Oluline informatsioon on minu poolt teile saadetud mailis.

ÜLESANNE

KOGU TEGEVUS (Sql-laused) tuleb protokollida ja protokoll esitada minule (paberil, lehed kokku klammerdatud).

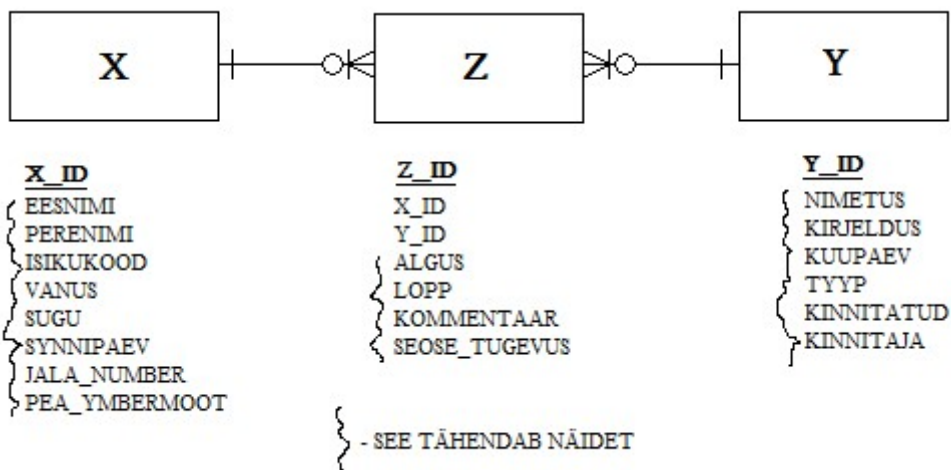
Igale SQL-lausele või lausete grupile peab eelnema 1-2 selgitavat lauset, mis kirjeldaks, mida te selle (või nende) SQL-lausega teete. Piisab sellest, kui lisate ülesande punktid nende lausete ette, millega te konkreetsetes punktis antud ülesannet täidate.

Pärast tabelite tegemist ja kirjade lisamist peate päringu (SELECT-lause) tulemusega kinnitama, et kõik see mis tegite on baasis olema. Tegelikult võiks iga punkti lõppedes, mis muudab baasi sisu, teha SELECT-lause ja selle väljundi lisada protokollile.

Protokollis peavad näha olema ka kõik SELECT-lausete päringute tulemused lausest endast üksi ei piisa.

Vaadake nädala 26.10.-06.11 loengute ja harjutustundide salvestusi ka. Seal on palju kasulikku selle ülesande tegemiseks.

1. Luua mingi kolmest tabelist koosnev andmemudel (tabelitele mõelge välja semantika ja pange korralikud nimed):



OLGE KENAD JA ÄRGE TEHKE TABELEID X, Y ja Z. Need on ainult näidis-olemite nimed.
Mõelge ikka välja mingi konkreetse elulise semantikaga tabelid.

JA SEE PEAB LOOGILISELT KORREKTNHE OLEMA! Näiteks tabelite nimed KÄGU-SIGA-ROTT ei sobi, sest need ei moodusta mingit kasutatavat mudelit. Kokku peavad nad moodustama mudeli mingist kahest objekt-tüüpi olemist ja nende vahelistest seose olemist. Kes sellega hakkama ei saa, sellel tuleb „ueele ringile“ minna, sest kogu muu tehtud töö muutub mõttetuks.

2. Teha SQL-laused selle skeemi loomiseks ORACLE andmebaasi:

- CREATE TABLE (3tk) – kirjeldage siis siin kohe PRIMARY KEY kah ära – on lihtsam. Kui tegemata ununeb, tuleb pärast ALTER TABLE-käsuga lisaks juurde teha
- Kirjeldage tabelile (Z) kaks FOREIGN KEY-d (ALTER TABLE- käsuga). Üks neist viitab tabelile X, teine tabelile Y.

(ärge unustage tegevust COMMIT-ida, kui õnnestus või siis ROLLBACK-ida, kui asi nässu läks; siis peab muidugi uuesti tegema)

3. Täida tabelid andmetega: tabelisse X pane 8 rida, tabelisse Z 15 rida ja tabelisse Y 9 rida. Read peavad omavahel andmete abil ka ilusasti seotud olema (võib ka koledasti aga seos peab olema :-).

(ärge unustage tegevust COMMIT-ida, kui õnnestus või siis ROLLBACK-ida, kui asi nässu läks; siis peab muidugi uuesti tegema)

4. Muutke tabelis olevate kirjete andmeid. Kõik muudatused võib teha ühes tabelis aga võib iga muudatuse teha ka erinevas tabelis:

- a. Muutke andmeid ühes kirjes (kirje valiku tingimus peab viitama ühele kirjele)
- b. Muutke andmeid mitmes kirjes korraga (kirjete valiku tingimus peab viitama mitmele kirjele)
- c. Muutke korraga andmeid ühe tabeli kõigis kirjetes (näiteks liitke mingile arv-veeru väärtusele 5; kirjete valiku tingimus peab viitama kõigile kirjetele või siis puuduma hoopis)

(ärge unustage tegevust COMMIT-ida, kui õnnestus või siis ROLLBACK-ida, kui asi nässu läks; siis peab muidugi uuesti tegema)

5. Kustutage tabelitest kirjeid:

- a. Ühest tabelist üks kirje (kirje valiku tingimus peab viitama ühele kirele)
- b. Ühest tabelist mitu kirjet (kirjete valiku tingimus peab viitama mitmele kirjele)
- c. Ühest tabelist kõik kirjed (kirjete valiku tingimus peab viitama kõigile kirjetele või siis puuduma hoopis)

(ärge unustage tegevust COMMIT-ida, kui õnnestus või siis ROLLBACK-ida, kui asi nässu läks; siis peab muidugi uuesti tegema)

6. Kuna teil on ettenägelikult alles need INSERT-laused, millega te kustutatud kirjed eelnevalt tabelitesse kirjutasite, siis taastage tabelitest kustutatud kirjed.

(ärge unustage tegevust COMMIT-ida, kui õnnestus või siis ROLLBACK-ida, kui asi nässu läks; siis peab muidugi uuesti tegema)

7. Tehke päring ühte tabelisse. Proovige erinevaid piiravaid tingimusi (WHERE) ja erinevaid kirjete järjestusi (ORDER BY). Proovige ka kõikide veergude väljastust (*) ja ka valikulist veergude väljastust (veergude loend üle koma)

(Nüüd pole COMMIT-i ega ROLLBACK-i vaja, sest andmebaasis ei muudetud midagi – ainult päriti)

8. Proovige leida üle väljade maksimume (MAX(...)), miinimume (MIN(...)) ja loendusandmeid (COUNT (...)). Seda nii üle terve tabeli (kõik read) ja ka ainult üle mõne kirje, mis valitakse välja piirangutega WHERE-tingimuses.

9. Tehke päring üle KAHE tabeli:

- a. Esialgu nii, et väljastatakse piiranguteta kõik read („piiranguta“ ei tähenda cartesiuse loendit st tabelite vahelised seosed peavad ikka kirjeldatud olema)
- b. Lisage piirang nii et väljastataks üks rida
- c. Muutke piirangut selliselt, et väljastataks mitu rida, aga mitte kõiki ridu.

!!! Pärast iga päringut (a, b ja c) tehke kontrollpäring sama tingimusega, mis eelmine päring ja funktsiooniga COUNT(), et kontrollida, kas eelmisest päringust tulnud ridade arv on sama, mis COUNT() päringust tulnud number.

10. Tehke päring üle KOLME tabeli:

- a. Esialgu nii, et väljastatakse piiranguteta kõik read („piiranguta“ ei tähenda cartesiuse loendit st tabelite vahelised seosed peavad ikka kirjeldatud olema)
- b. Lisage piirang nii et väljastataks üks rida
- c. Muutke piirangut selliselt, et väljastataks mitu rida, aga mitte kõiki ridu.

!!! Pärast iga päringut (a, b ja c) tehke kontrollpäring sama tingimusega, mis eelmine päring ja funktsiooniga COUNT(), et kontrollida, kas eelmisest päringust tulnud ridade arv on sama, mis COUNT() päringust tulnud number.

VEEL KORD: KOGU TEGEVUS (Sql-laused) tuleb protokollida ja protokoll mulle esitada. Iga SQL-lausele peab eelnema 1-2 lauset, mis selgitaks, mida selle SQL-lausega „öelda on tahetud“. Need laused võib võtta ülesandest.

Iga SQL-lause kohta peab olema protokollis näha ka lause täitmise tulemus. Näiteks CRETE TABLE täitmise järel peab olema veel SELECT lause (SELECT * FROM tabeli_nimi), mille tulemus näitab, et tabeli loomine tõesti õnnestus.

SELECT-lause täitmise tulemusena peavad olema näha need read, mille päring välja andis.