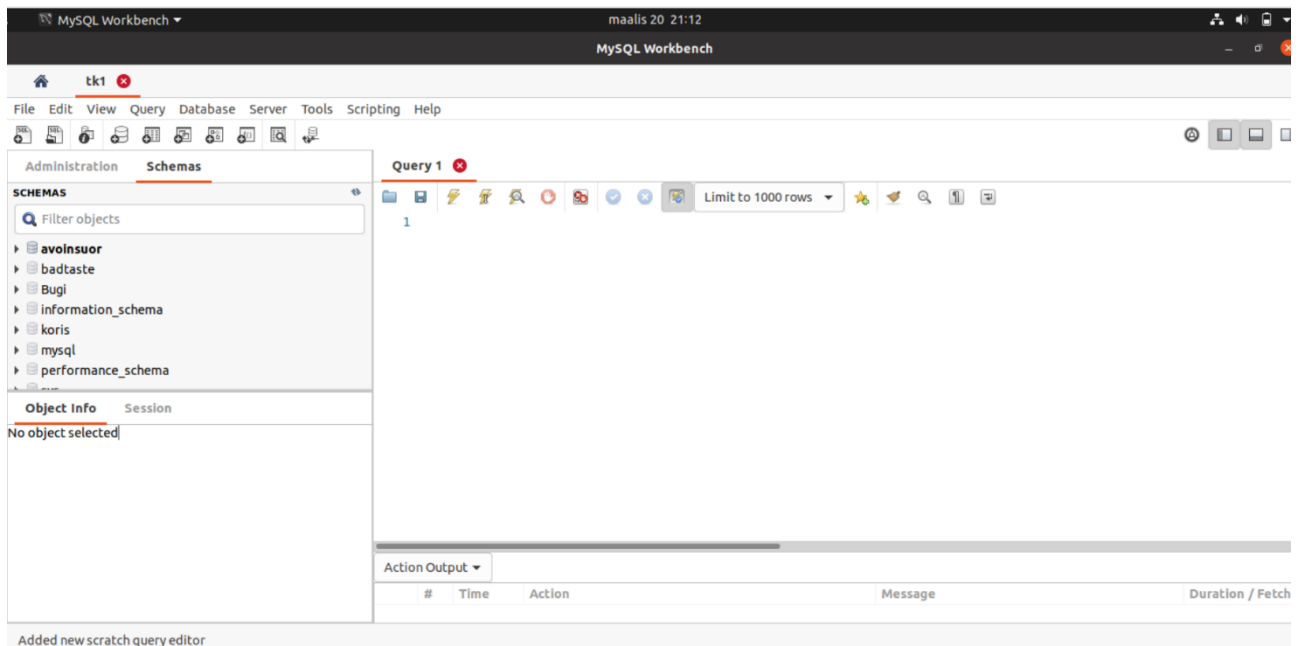


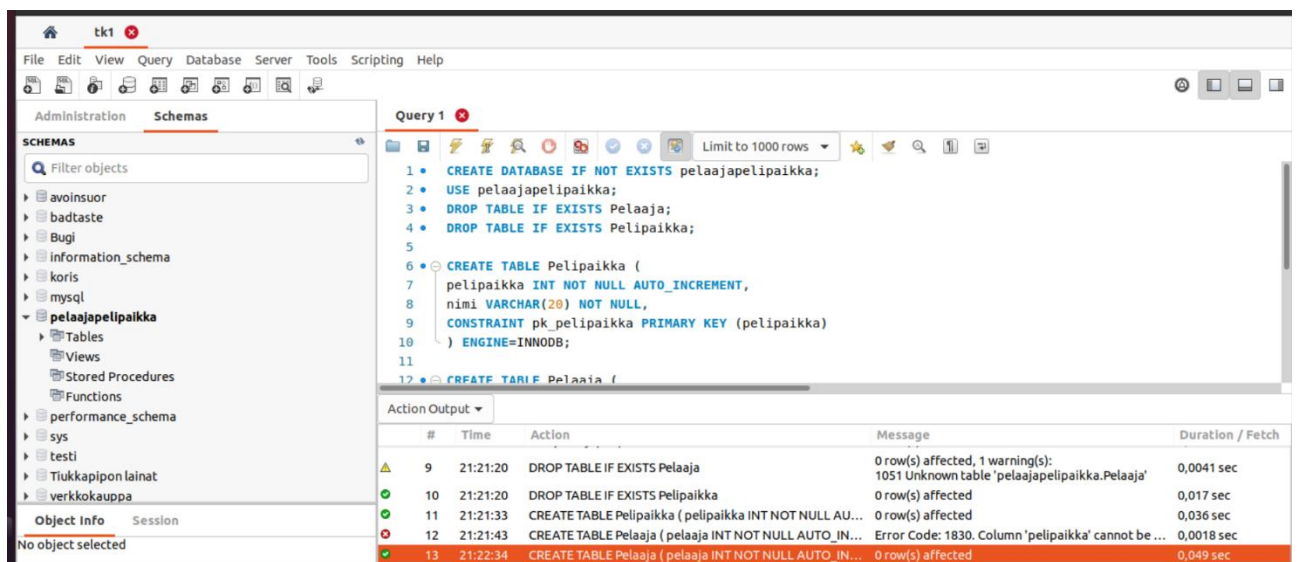
Kotitehtävät, relaatiokannat

1. Tehtävä

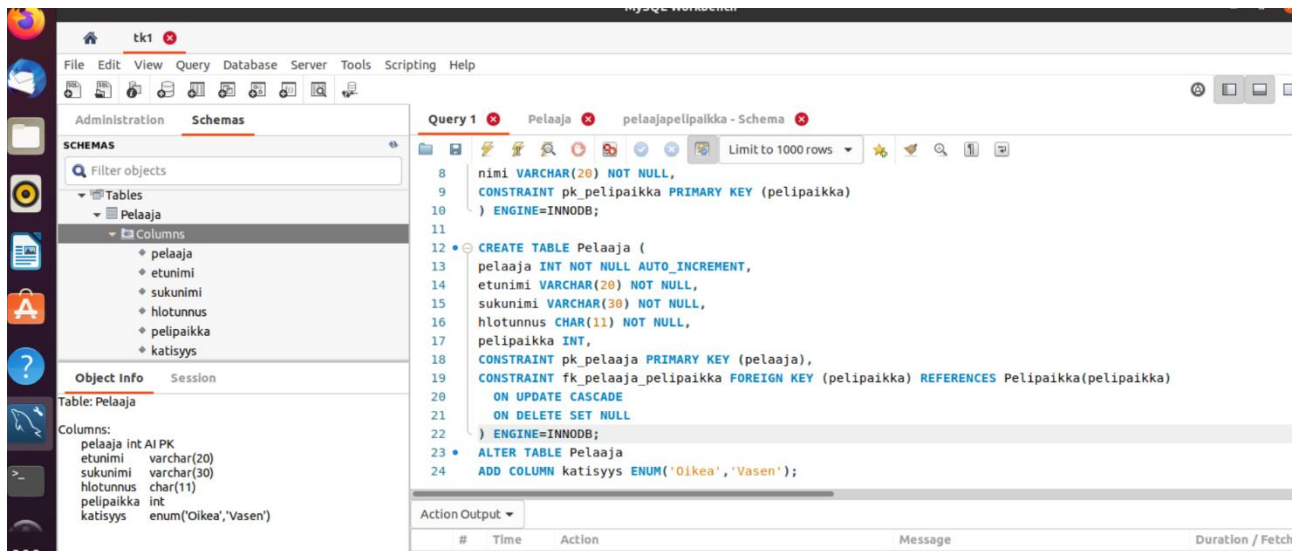


2. Tehtävä

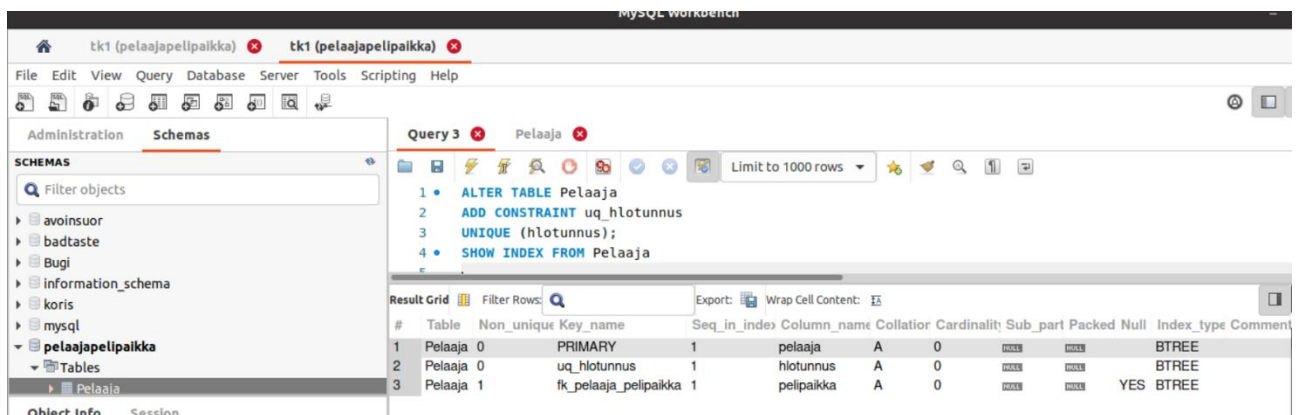
Mieleenpalautusta vanhoista asioista: oheisen skriptin pitäisi luoda kanta ja kaksi taulua, mutta se ei toimi. Etsi virhe ja korjaa se.



- Määrittele pelaajan kätisyys luettuna tyyppinä, jossa mahdolliset arvot ovat vain "Oikea" ja "Vasen". Tämä on uusi asia, joten lue aluksi luettelusta tyyppistä.



- Määrittele hlotunnus-kentälle yksikäsitteisyys-rajoite. Nimeä se rajoitenimellä "uq_hlotunnus". Lisäinfoa.



3. Tehtävä

Liityt projektiin, jossa ollaan suunnittelemassa tilausohjelmistoa. Projektissa ei ole juurikaan kantaosaamista, joka voi helposti havaita esimerkiksi alla olevasta TILAUS-taulusta, joka ei ole edes ensimmäisessä normaalimuodossa. Normalisoi ratkaisu aluksi 1NM, sitten 2NM ja lopuksi 3NM. Esitä kunkin vaiheen malli oheisen datan avulla käyttäen esimerkiksi taulukkolaskentaohjelmistoa. Tässä tehtävässä on tarkoitus edetä systemaattisesti, älä siis "etuile" vaikka arvaisitkin lopputuloksen.

1NM

tilausnro	pvm	asiakas_nro	asiakas	tuote_nro	tuote	hinta	määrä
100	1.2.2021	1	Antti	VA100	Piimä	1,5	3
100	1.2.2021	1	Antti	VA101	Maito	1	2
100	1.2.2021	1	Antti	VA102	Maitorahka	2	1
101	2.2.2021	2	Bertta	VA101	Maito	1	10

2NM

Tilaus

tilausnro	pvm	asiakas_nro	tuote_nro	määrä
100	1.2.2021	1	VA100	3
100	1.2.2021	1	VA101	2
100	1.2.2021	1	VA102	1
101	2.2.2021	2	VA101	10

Asiakas

asiakas_nro	asiakas
1	Antti
2	Bertta

Tuote

tuote_nro	tuote
VA100	Piimä
VA101	Maito
VA102	Maitorahka
VA101	Maito

3NM

Pvm

tilausnro	pvm
100	1.2.2021
100	1.2.2021
100	1.2.2021
101	2.2.2021

Tilaus

tilausnro	asiakas_nro	tuote_nro	määrä
100	1	VA100	3
100	1	VA101	2
100	1	VA102	1
101	2	VA101	10

Asiakas

asiakas_nro	asiakas
1	Antti
2	Bertta

Tuote

tuote_nro	tuote
VA100	Piimä
VA101	Maito
VA102	Maitorahka
VA101	Maito

4. Tehtävä

Tee aluksi kanta ohjeisesta skriptistä ja tutki sen rakennetta.

- a) Listaa henkilöt, heidän kotipaikkansa ja maantieteellinen alue (Helsinki -> Etelä-Suomi, Turku -> Länsi-Suomi, Tampere -> Häme, muut -> Muu Suomi). Määrittele maantieteellinen alue -sarakkeen nimeksi alue.

```
Execute:
> SELECT Henk.snimi, Henk.kunta, Area.alue FROM Henk
INNER JOIN Area ON Henk.kunta=Area.kunta

+-----+ +-----+ +-----+
| snimi  | | kunta  | | alue   |
+-----+ +-----+ +-----+
| Virta  | | TURKU  | | Lansi-Suomi |
| Metsä  | | TURKU  | | Lansi-Suomi |
| Joki   | | HELSINKI | | Etela-Suomi |
| Meri   | | TURKU  | | Lansi-Suomi |
| Järvi  | | HELSINKI | | Etela-Suomi |
| Ranta  | | TAMPERE | | Hame    |
+-----+ +-----+ +-----+
6 rows
```

- b) Listaa sellaisten henkilöiden nimet, joiden kotipaikka on TURKU tai HELSINKI. Nimen lisäksi kyselyn tulee palauttaa info-sarake, jossa helsinkiläisten osalta on palkka ja turkulaisten osalta ikä kyseisenä ajanhetkenä (älä käytä vakioarvoa kuluvalle vuodelle vaan DBMS:n tarjoamia muita keinoja). Info-sarakkeessa tulee olla myös tieto, että mitä tietoa sarake sisältää (esimerkiksi "Ikä: 59"). Pyöristä palkat kokonaisluvuiksi.

```
Execute:
> SELECT Henk.snimi,
IF (kunta='TURKU',(YEAR(CURDATE())-Henk.svuosi),ROUND(palkka,-1)) AS info
FROM Henk
```

snimi	info
Virta	57
Metsä	46
Joki	6230
Meri	38
Järvi	6100
Ranta	2500

6 rows

5.Tehtävä

Tee edelliseen prokantaan seuraavat toimenpiteet:

- a) Tee alikyselyä käyttäen kysely, joka palauttaa htun-, sukunimi- ja svuosi-sarakkeet sellaisista henkilöistä, jotka ovat nuorempia tai yhtä vanhoja kuin Ranta (htun: 2983). Tee sitten sama kysely ilman alikyselyä käyttäen CROSS JOIN -liitosta suurempi kuin -ehdolla.

```
Execute:
> SELECT htun, snimi, svuosi
FROM prokanta.Henk
WHERE
svuosi > (
SELECT svuosi FROM prokanta.Henk
WHERE htun='2983'
)
OR
svuosi = (
SELECT svuosi FROM prokanta.Henk
WHERE htun='2983'
)
```

htun	snimi	svuosi
2234	Metsä	1976
2245	Joki	1974
2345	Meri	1984
2983	Ranta	1974
NULL	NULL	NULL

5 rows

```
Execute:
> SELECT y.htun, y.snimi, y.svuosi
FROM Henk x
CROSS JOIN Henk y ON y.svuosi>x.svuosi OR y.svuosi=x.svuosi
WHERE x.htun = 2983
```

htun	snimi	svuosi
2234	Metsä	1976
2245	Joki	1974
2345	Meri	1984
2983	Ranta	1974

4 rows

- b) Hae samaan tulosjoukkoon sijaintipaikoittain kaikki henkilöt ja projektit (sijaintipaikka/kunta, htun/ptun). Rivillä pitää näkyä, onko kyseessä henkilö vai projekti. Järjestä sijaintipaikoittain.

```
Execute:
> SELECT kunta, htun,
CASE
WHEN htun THEN 'HENKILO'
ELSE 'PROJEKTI'
END AS info
FROM prokanta.Henk
UNION ALL
SELECT sijainti, ptun, 'PROJEKTI'
FROM prokanta.Proj
ORDER BY kunta ASC
```

kunta	htun	info
HELSINKI	P6	PROJEKTI
HELSINKI	2245	HENKILO
HELSINKI	2884	HENKILO
HELSINKI	P2	PROJEKTI
KUOPIO	P5	PROJEKTI
OULU	P3	PROJEKTI
TAMPERE	2983	HENKILO
TURKU	2134	HENKILO
TURKU	2234	HENKILO
TURKU	2345	HENKILO
TURKU	P1	PROJEKTI
TURKU	P4	PROJEKTI

6. Tehtävä

Tee edelleen prokantaan seuraavat toimenpiteet:

- a) Kirjoita kysely, joka listaa henkilöiden keskipalkat kaupungittain. Pöyristä kokonaislukuun. Saat kaiken tarvittavan tiedon Henk-taulusta.

```
Execute:
> SELECT kunta, ROUND(AVG(palkka),0) FROM prokanta.Henk
WHERE kunta='TURKU'
UNION ALL
SELECT kunta, ROUND(AVG(palkka),0) FROM prokanta.Henk
WHERE kunta='HELSINKI'
UNION ALL
SELECT kunta, ROUND(AVG(palkka),0) FROM prokanta.Henk
WHERE kunta='TAMPERE'
```

kunta	ROUND(AVG(palkka),0)
TURKU	4777
HELSINKI	6165
TAMPERE	2500

3 rows

b) Tee edellisestä kyselystä näkymä PalkkaKeskiarvot.

Filter objects

prokanta

Tables

Area

Henk

Prhe

Proj

Views

PalkkaKeskiarvot

Stored Procedures

Object Info Session

View: PalkkaKeskiarvot

Columns:

1 • SELECT * FROM prokanta.PalkkaKeskiarvot;

Result Grid

Filter Rows

Export: Wrap

#	kunta	ROUND(AVG(palkka),0)
1	TURKU	4777
2	HELSINKI	6165
3	TAMPERE	2500

PalkkaKeskiarvot 1

7. Tehtävä

Kopioi prokannan Henk-taulun rakenne (ilman sisältöä) tauluksi UudetHenk. Käytä jotain materiaalissa esitettyä tapaa, älä luettele sarakkeita "normaaliin" tapaan.

Kirjoita triggeri, joka aktivoituu kun Henk-tauluun lisätään uusi rivi. Triggeri lisää Henk-tauluun lisätyn henkilön tiedot myös UudetHenk-tauluun.

Testaa toimintaa lisäämällä Henk-tauluun henkilö, jonka käyttäjätunnus on 9876, sukunimi Doe, syntymävuosi 2002 ja palkka 990.

```
CREATE TRIGGER newPerson AFTER INSERT ON Henk
```

```
FOR EACH ROW
```

```
INSERT INTO UudetHenk (htun, snimi, svuosi, palkka)
```

```
VALUES (NEW.htun, NEW.snimi, NEW.svuosi, NEW.palkka);
```

```
INSERT INTO Henk (htun, snimi, svuosi, palkka) VALUES (9876,'Doe', 2002, 990);
```

```
Execute:
> SELECT * FROM prokanta.UudetHenk
```

htun	snimi	kunta	svuosi	palkka
9876	Doe	2002	990.0000	
NULL	NULL	NULL	NULL	NULL

2 rows

8. Tehtävä

Lisää prokantaan oheinen Ylittajat-taulu. Kirjoita sen jälkeen triggeri, joka aktivoituu kun Prhe-tiluun on tehty päivitys. Triggeri tutkii, että ylittääkö henkilön tietyille projektille tekemien tuntien määrä (tunnit) suunnittelun määrän (tunnit_suun) päivityksen jälkeen ja lisää Ylittajat-tiluun riviin (htun, ptun ja sen hetkinen päivämäärä) mikäli näin on. Lisää triggerin ehtoon myös tarkastus, että Ylittajat-tiluun ei lisätä riviä, mikäli tuntien määrä oli suurempi kuin suunniteltu määrä jo ennen päivitystä. Mikäli tunnit eivät ole ylittäneet suunniteltua määrää, triggeri ei tee mitään.

```
CREATE TABLE Ylittajat(
id INT PRIMARY KEY AUTO_INCREMENT,
htun CHAR(4) NOT NULL,
ptun CHAR(4) NOT NULL,
pvm DATE NOT NULL
) ENGINE=INNODB;

DROP TRIGGER ylittaja
DELIMITER //
CREATE TRIGGER ylittaja AFTER UPDATE ON Prhe
FOR EACH ROW
BEGIN
    IF NEW.tunnit > OLD.tunnit_suun THEN
        INSERT INTO Ylittajat (htun, ptun, pvm)
        VALUES (OLD.htun, OLD.ptun, NOW());
    END IF;
END;
//
```

9. Tehtävä

Prokannan Henk-tilussa ihmisten kotikunnat on kirjoitettu kokonaan suuraakkosilla. Kirjoita funktio kuntaNormaalinaTekstina, joka saa parametrikseen kunnan nimen (nimeä parametri p_kunta:ksi) ja joka palauttaa kunnan nimen "normaalisti" kirjoitettuna. Esimerkiksi "HELSINKI" palautuu funktiosta muodossa "Helsinki".

Kirjoita funktion toiminnan testaamiseksi SELECT-lause, joka tulostaa kaikkien henkilöiden sukunimet sekä heidän kuntansa kannasta että funktion läpi ajattuna.

```
DELIMITER //

CREATE FUNCTION kuntaNormaalinaTekstina(p_kunta CHAR(10)) RETURNS CHAR(10)
DETERMINISTIC CONTAINS SQL

RETURN

    CONCAT(
```

```
LEFT(p_kunta,1),  
LOWER(SUBSTR(p_kunta FROM 2))  
);  
//
```

10. Tehtävä

Kirjoita prokantaan funktio `palkkaArvio`, joka saa parametrinaan henkilön tunnuksen (`htun`-sarake `Henk`-taulussa). Funktio selvittää kaikkien `Henk`-taulun ihmisten keskipalkan ja palauttaa sen pohjalta merkkijonon, joka kuvaa henkilön palkan suuruutta.

Palkan suuruus arvioidaan seuraavasti:

Palkka on maksimissaan 50% kaikkien palkkojen keskiarvosta -> "Palkkakuoppa"

Palkka on enemmän kuin palkkakuopan raja, mutta alle keskiarvon -> "Alle keskipalkan"

Palkka on vähintään keskiarvon verran, mutta alle keskiarvo + keskiarvon puolikas -> "Yli keskipalkan"

Palkka on enemmän kuin edellisellä -> "Roope Ankka"

Arviotekstin asettamiseen lienee järkevintä käyttää `CASE`-lauseketta. Huomaa, että `CASE`-lauseke on hieman erilainen kuin aiemmin käsitelty `CASE`-funktio. Voit asettaa paikallisiin muuttujiin (`DECLARE`-avainsanalla määritellyt muuttujat, esimerkiksi `DECLARE arvio VARCHAR(20)`) käyttämällä `SET`-avainsanaa, esimerkiksi `SET arvio = 'Roope Ankka'`.

Tee kysely, joka näyttää kaikki `Henk`-taulun ihmisten sukunimet, palkat ja palkka-arvion. Alla esimerkki tulosteesta, arvot voivat olla erilaisia kuin omassa kannassasi.