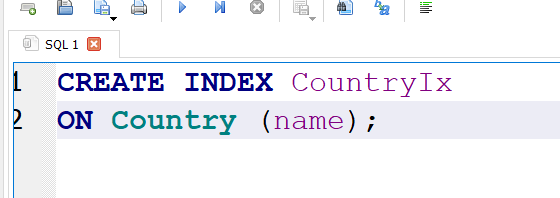
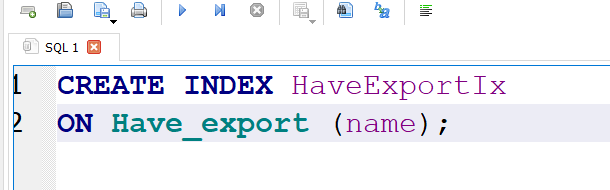
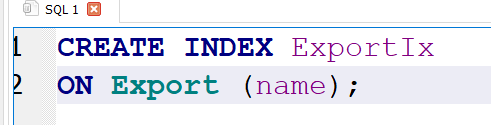
Акостакіоає Флоріан 414 КН

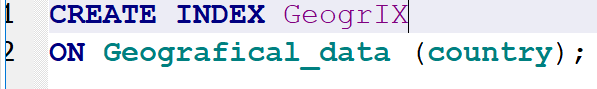
Білет 9

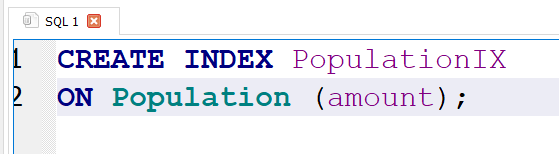
1. 2) Group by
2. 1) CREATE TRIGGER
3. 4) Виведуться всі записи з таблиці my\_table
4. 4) Для управління користувачами
5. 2) Впорядковує записи по зростанню
6. 3)1)4)
7. 2) Постійно зберігає вибірку даних
8. 3)
9. 3)
10. 1) 3) 5)
11. 3) 4) 5) 6)











17. 1

SELECT

Country.name,

Geografical\_data.area

FROM Country INNER JOIN Geografical\_data

ON Geografical\_data.country = Country.name

WHERE Country.name IN

(

SELECT Have\_export.country

FROM Have\_export

WHERE Have\_export.name LIKE "%metal%" OR

Have\_export.name LIKE "%himia%"

)

17. 2

SELECT

Country.name

FROM Country INNER JOIN Have\_export

ON Country.name = Have\_export.country

WHERE Country.continennt != "Africa" AND Have\_export.name = "coffe"

17. 3

SELECT

Country.name

FROM Country INNER JOIN Have\_export

ON Country.name = Have\_export.country

WHERE Country.continennt = "Eurasia" AND Have\_export.name = "wood"

18. 1

CREATE VIEW "CG\_prod" AS

SELECT

Country.name

FROM Country INNER JOIN Have\_export

ON Country.name = Have\_export.country

WHERE Have\_export.name NOT LIKE "%CG\_prod%"

18. 2

CREATE VIEW "ExportGoods" AS

SELECT

Country.name,

COUNT(Have\_export.country)

FROM Country INNER JOIN Have\_export

ON Country.name = Have\_export.country

GROUP BY Country.name

18. 3

CREATE VIEW AvgAfrica AS

SELECT

Country.name,

AVG(Population.VVP)

FROM Country INNER JOIN Population

ON Country.name = Population.country

GROUP BY Country.name

HAVING Country.name = "Africa"

19. 1

SELECT

(

SELECT AVG(VVP) FROM Population

INNER JOIN Country ON

Population.country = Country.name

WHERE Country.continennt = "Eurasia"

) AS AveragePerEurasia,

Country.name

FROM Country INNER JOIN Population

ON Country.name = Population.country

WHERE Country.continennt = "Eurasia" AND Population.VVP < AveragePerEurasia

19 . 2

SELECT

(

SELECT AVG(VVP) FROM Population

) AS AveragePerWorld,

Country.name

FROM Country INNER JOIN Population

ON Country.name = Population.country

WHERE Population.VVP > AveragePerWorld

19. 3

SELECT

Have\_export.name

FROM Have\_export

WHERE Have\_export.country IN

(

SELECT

Country.name

FROM Country INNER JOIN Geografical\_data

ON country.name = Geografical\_data.country

WHERE Geografical\_data.area < 1000000

)

BEGIN TRANSACTION;

INSERT INTO Economical\_data

(money) VALUES ("grivna");

INSERT INTO Languages

(language) VALUES ("Uknrainian");

INSERT INTO Relief

(relief) VALUES ("Good relief I like it");

INSERT INTO Export

(name) VALUES ("Some salo");

DELETE FROM Export

WHERE name = "Some salo";

UPDATE Country

SET second\_city = "Chernivtsy"

WHERE Country.name = "Ukraine";

COMMIT;

BEGIN TRANSACTION;

CREATE TABLE "Country"

(

"name" VARCHAR NOT NULL,

"money" VARCHAR NOT NULL,

"government\_type" VARCHAR NOT NULL,

"capital" VARCHAR NOT NULL,

"second\_city" VARCHAR,

"continennt" VARCHAR NOT NULL,

FOREIGN KEY (money) REFERENCES Economical\_data(money)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (government\_type) REFERENCES Political\_data(government\_type)

ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (continennt) REFERENCES Continent(name)

ON DELETE CASCADE ON UPDATE CASCADE , PRIMARY KEY("name")

);

CREATE TABLE Have\_relief ( relief VARCHAR NOT NULL, country VARCHAR NOT NULL, FOREIGN KEY (relief) REFERENCES Relief(relief) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (country) REFERENCES Country(name) ON DELETE CASCADE ON UPDATE CASCADE )

CREATE TABLE Languages ( language VARCHAR NOT NULL, PRIMARY KEY("language") )

CREATE TABLE Political\_data ( government\_type VARCHAR NOT NULL, PRIMARY KEY("government\_type") )

CREATE TABLE Population ( amount INT NOT NULL, VVP VARCHAR NOT NULL, specialty VARCHAR NOT NULL, country VARCHAR NOT NULL, FOREIGN KEY (country) REFERENCES Country(name) ON DELETE CASCADE ON UPDATE CASCADE )

CREATE TABLE Relief ( relief TEXT, PRIMARY KEY("relief") )

COMMIT;