UNIVERSITETI I PRISHTINËS ''HASAN PRISHTINA'' FAKULETI I SHKENCAVE MATEMATIKE-NATYRORE Departamenti: Matematikë | Programi: Shkencë kompjuterike

Janar 2023

Bazat e të dhënave

Online news website with user preferences

Raport i projektit

Punuar nga:

Diellza Bajraktari

Definimi i projektit

Për faqen e lajmeve online me projektin e bazës së të dhënave të preferencave të përdoruesve, ne filluam duke analizuar kërkesat për bazën e të dhënave dhe duke përcaktuar llojet e të dhënave që duheshin ruajtur. Bazuar në këtë analizë, ne krijuam një bazë të dhënash me shtatë tabela: Users, Categories, Article ,User Preferences, User Article History, Article Ratings, and Comments.

Për t'u siguruar që baza e të dhënave ka ndjekur rregullat e integritetit, ne kemi implementuar çelësat primarë dhe çelësat e huaj në tabela për të zbatuar marrëdhëniet midis të dhënave.

Ne përdorëm gjithashtu lloje të përshtatshme të të dhënave dhe kufizime për kolonat në tabela, si p.sh. përdorimi i llojit të të dhënave VARCHAR për fushat e tekstit dhe lloji i të dhënave INTEGER për fushat numerike.

Qëllimi i projektit

Qëllimi i një portali lajmesh në internet me preferencat e përdoruesve është t'u ofrojë përdoruesve akses në një gamë të gjerë lajmesh dhe informacionesh dhe t'i lejojë ata të personalizojnë përvojën e tyre duke zgjedhur temat, burimet dhe shpeshtësinë e përditësimeve që janë më të rëndësishme për interest e tyre.

Tabelat:

Users: Composite Primary Key(users\_Username,user\_Email), users\_Password, users\_Zip, users\_Street, users\_City, user\_State.

Articles: article\_ID(PK), article\_Title, article\_author, article\_publicationdate, article\_history, article\_rating, article\_content, article\_category

Categories: category\_id(PK), category\_name(PK)

User Preferences: userpref\_id(PK) userpref\_username( FK), user\_pref\_category\_id(FK)

User Article History: Composite Primary Key (userhistory\_username, userhistory\_article\_id) (FK).

Article Ratings: Composite Primary Key(articleratings\_username, article\_ratings\_id) (FK), article\_rating.

Comments: Comment\_id (PK), comment\_content, parent\_id (FK nga Comments), comments\_username(FK), comments\_articleid(FK).

Arkitektura

Ne figurën 1. kemi paraqitur ER diagramin :

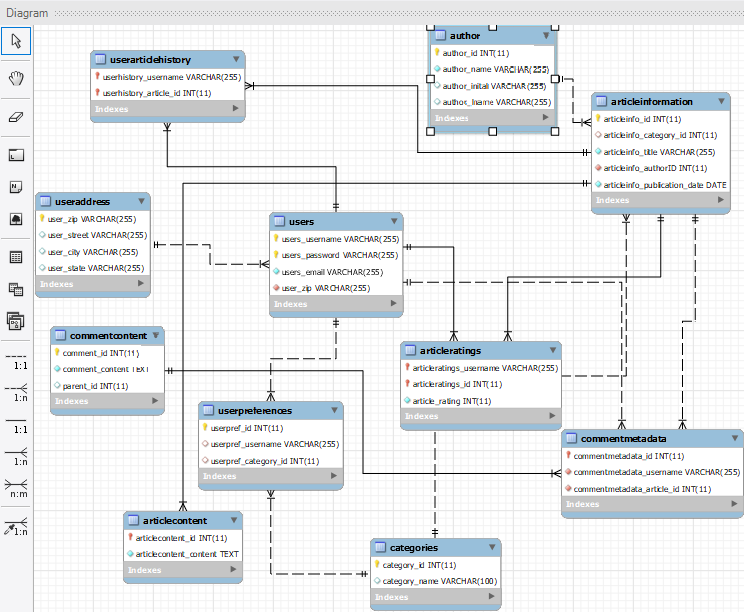


Figure 1: ER Diagrami per bazen e te dhenave

Crow’s Foot

Në këtë diagram është paraqitur marrëdhënia e klasave që kemi krijuar në MySQL. Kjo marrëdhenie tregon lidhjet mes entiteteve të cilat I kemi shprehur me fraza që përshkruajnë secilën prej atyre lidhjeve. Kardinaliteti që është përdorur në këtë diagram është kardinaliteti një me shumë me anë të simboleve.

Në figurën 2. kemi paraqitur Crow’s Foot diagramin :

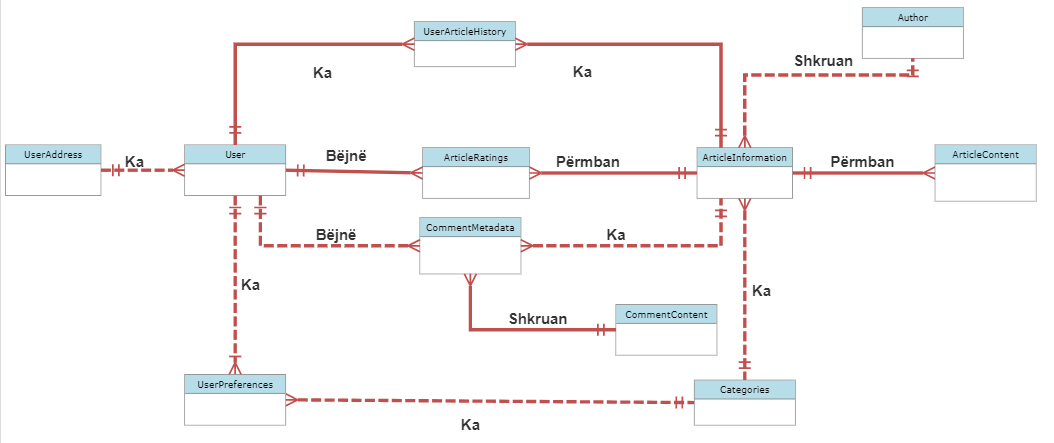


Figure 2.: Crow's Foot

UML Modeli Me modelin UML janë definuar lidhjet më specifike mes dy tabelave. Në këtë rast për lidhje janë përdorur indikatorët 1..1, 1..m. Në figurën 3. është paraqitur UML modeli

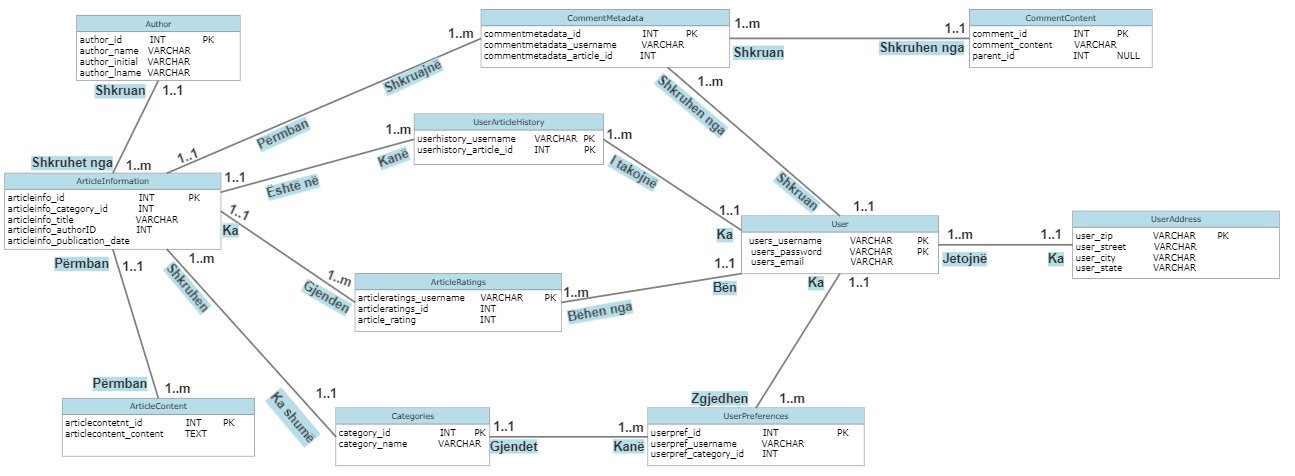


Figure 3: UML Class Model

Object-Oriented Modeli Në këtë diagram tregohet pamja e plote e databazës, ku objektet që kanë karakteristika të ngjajshme grupohen në klasa.Në figurën 4. është paraqitur Object-Oriented modeli.

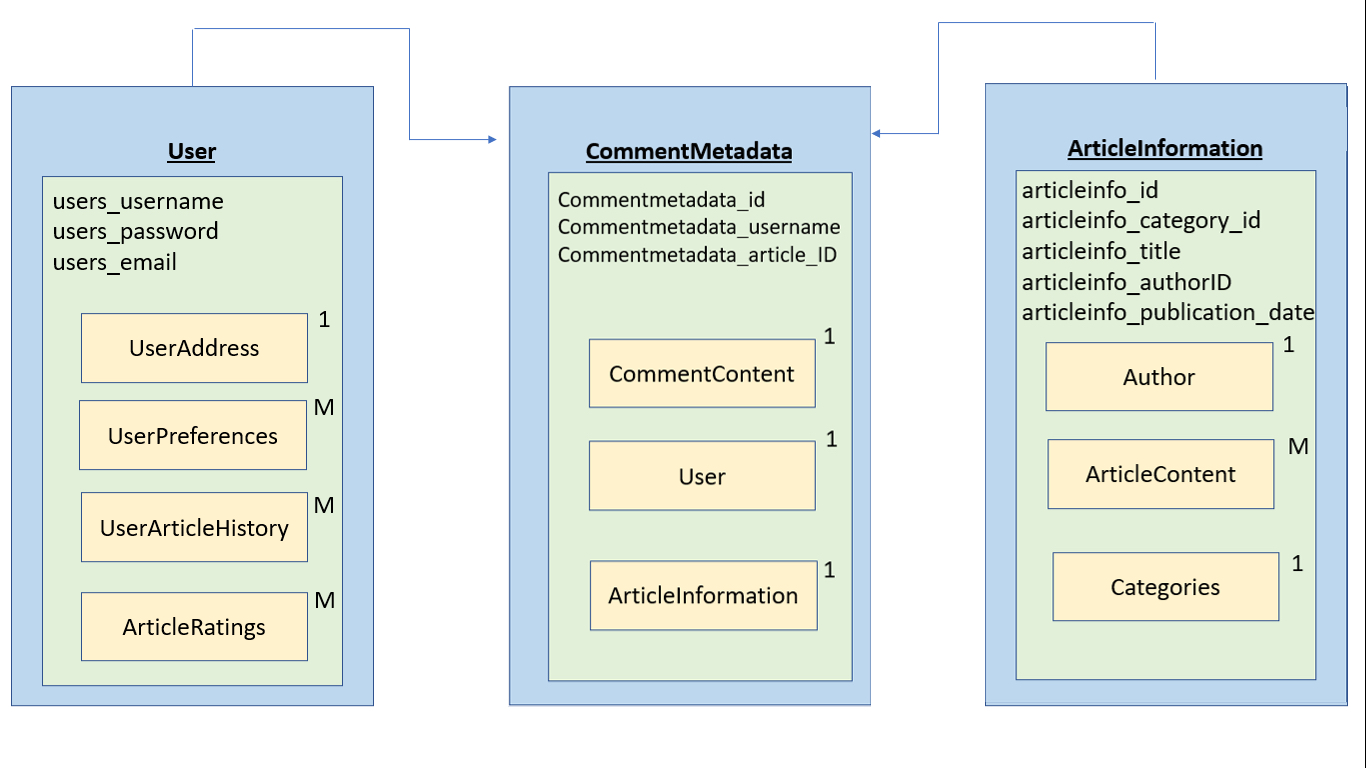


Figure 4: OO Modeli

## Normalizimi i bazës së të dhënave.

1. Tabela USERS përmban fushat users\_Username(PK), users\_Password, users\_Email(PK), users\_Zip, users\_Street, users\_City, user\_State. Për ta normalizuar këtë tabelë në 3NF duhet fillimisht t’a identifikojmë çelësin primarë dhe vërejmë që të gjitha atributet tjera varen nga çelësi primar prandaj është në 1NF. Mirëpo vërejmë se atributet p.sh users\_Zip, users\_Street, users\_City, user\_State varen vetëm nga një pjesë e çelësit primar users\_Username dhe jo nga users\_Email. Prandaj duhet t’i ndajmë atributet që varen vetëm nga një pjesë e çelësit primarë në një tabelë të veçantë dhe të krijojmë një lidhje midis dy tabelave.

Password

Street

**Email**

State

City

Zip

**Username**

Pra siç vërejmë nga diagrami më lartë përveç varësisë parciale kemi edhe lidhje transitive e cila është pasi që atributet Street, City, State varen nga atributi ZIP andaj kemi varësi tranzitive. Prandaj duhet t’i ndajmë atributet që varen vetëm nga një pjesë e çelësit primarë në një tabelë të veçantë dhe ato me varësi tranzitive në një tabelë të veçantë dhe të krijojmë një lidhje midis dy tabelave.

Prandaj do te kemi

Tabela 1 (USERS): users\_Username(PK), users\_Password, users\_Email(PK), users\_zip(FK)

Tabela 2(USER\_ADDRESS): users\_Zip(PK), users\_Street, users\_City, users\_State

1. Tabela Articles

Tabela Articles ka atributet article\_ID(PK), article\_Title, article\_author, article\_publicationdate, article\_content, article\_category. Verejme që secili atribut varet nga çelësi primar. Poashtu meqë ka vetëm një çelës primar nuk ka varësi parciale. Na mbetet t’i eleminojmë varësitë tranzitive.

**Article\_ID** Article\_Title A\_Author A\_PublicDate A\_Content A\_Category

Prandaj pasi e normalizojmë në 3NF kemi:

Tabela 1 ArticleInformation: articleinfo\_id(PK), articleinfo\_categoryID(FK nga tabela Categories), articleinfo\_title, articleinfo\_authorID(FK nga tabela Categories), articleinfo\_publicationdate

Tabela 2 ArticleContent: articlecontent\_id(PK, FK nga tabela ArticleInformation), articlecontent\_content.

Tabela 5 Author: author\_id(PK), author\_fname, author\_intital, author\_lname.

1. Tabela Categories

**category\_id**

category\_name

Qartazi vërejmë se tabela Categories është në 3NF.

1. Tabela User Preferences

**User\_pref\_id**

Userpref\_username

Userpref\_categoryID

Tabela është në formën e normalizuar 3NF.

1. Tabela UserArticleHistory

**userhistory\_articleID**

**userhistory\_username**

Kjo tabelë është një entitet i përbërë i krijuar vetëm për të mundësuar lidhjen shumë me shumë në mes të Articles dhe Users

1. Tabela ArticleRatings

**article\_\_ratings\_id**

**username**

article\_rating

Edhe tabela ArticleRatings është në 3NF.

1. Tabela Comments

**Comment\_id** C\_content parent\_id c\_username c\_articleid

Pra kemi varësi tranzitive pasi c\_username varet nga c\_content prandaj krijojmë tabelat në vijim :

Tabela 1 (CommentContent) : comment\_id(PK), comment\_content, parent\_id.

Tabela 2 (CommentMetaData) : commentmetadata\_id(PK, FK nga CommentContent), commentmetadata\_username(FK nga Users), commentmetadata\_article\_id(FK nga ArticleInformation).

## SQL Queries

1. Për të nxerrur një listë artikujsh për një përdorues bazuar në preferencat e tyre

SELECT users\_username, articleinfo\_title

FROM Users

JOIN UserPreferences ON users\_username = userpref\_username

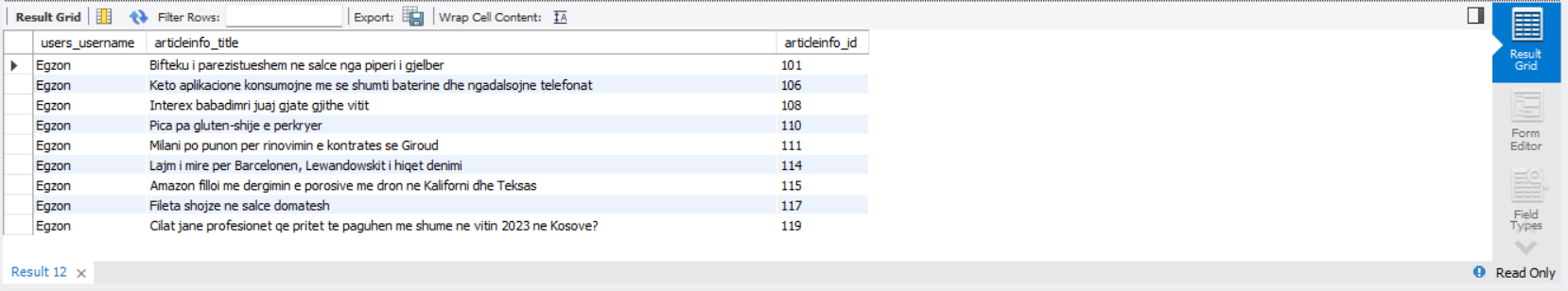
JOIN articleinformation ON userpref\_category\_id = articleinfo\_category\_id

WHERE users\_username = 'Egzon';

ORDER BY articleinfo\_id

Algjebër relacionale :

π(username, article\_title, article\_id) (σ(username = [username])(Users) × UserPreferences × Articles)

Rezultati:

1. Për të nxerrur historinë e artikujve të një përdoruesi:

SELECT users\_username, userhistory\_article\_id, articleinfo\_category\_id, articleinfo\_title

FROM Users

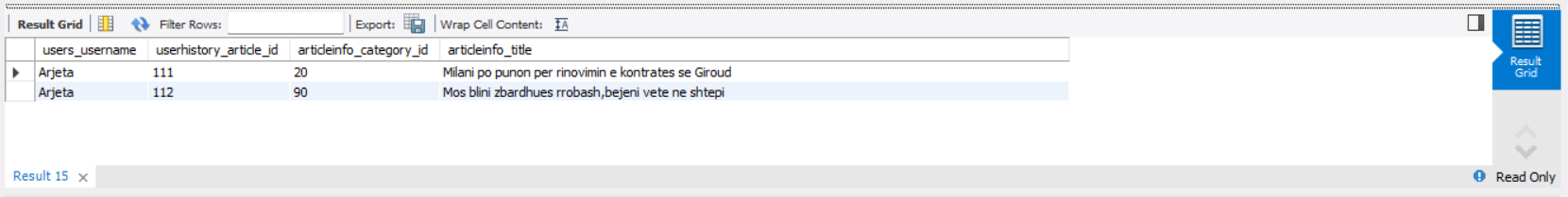
JOIN userarticlehistory ON users\_username = userhistory\_username

JOIN articleinformation ON userhistory\_article\_id = articleinfo\_id

WHERE users\_username = 'Arjeta';

Algjebër relacionale:

π(username, article\_id, category\_id, article\_title) (σ(username = [username])(Users) × UserHistory × Articles)



1. Për të marrë vlerësimin mesatar për një artikull:

SELECT AVG(article\_rating)

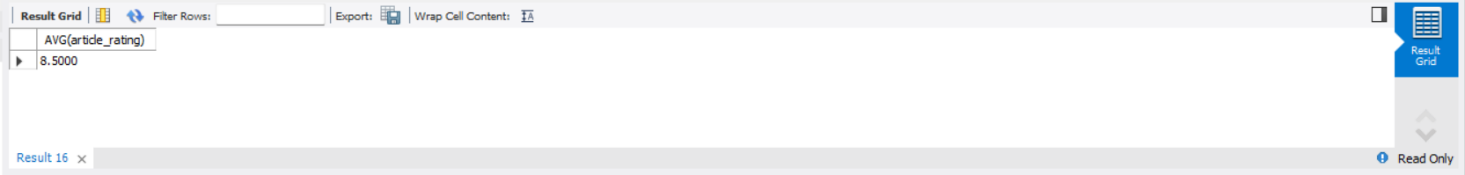
FROM ArticleRatings

WHERE articleratings\_id = '119';

Algjebër relacionale:

AVG(σ(article\_id = [article\_id])(ArticleRatings))

Rezultati:



1. Për të nxerrur artikujt më të njohur (bazuar në numrin e shikimeve):

SELECT articleinfo\_id, articleinfo\_category\_id, articleinfo\_title,

COUNT(userhistory\_article\_id) AS views

FROM articleinformation

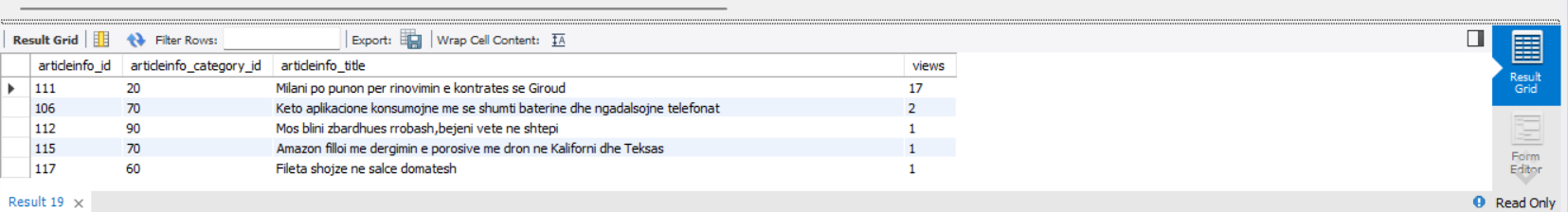
JOIN userarticlehistory ON articleinfo\_id = userhistory\_article\_id

GROUP BY articleinfo\_id

ORDER BY views DESC;

Algjebër relacionale:

π(articleinfo\_id, articleinfo\_category\_id, articleinfo\_title, COUNT(userhistory\_article\_id) as views) (σ(articleinfo\_id = userhistory\_article\_id) (articleinformation ⨝ userarticlehistory))



1. Për të marrë artikujt më të vlerësuar (bazuar në vlerësimin mesatar):

SELECT articleinfo\_id, articleinfo\_category\_id, articleinfo\_authorID,

AVG(article\_rating) AS average\_rating

FROM articleinformation

JOIN articleratings ON articleinfo\_id = articleratings\_id

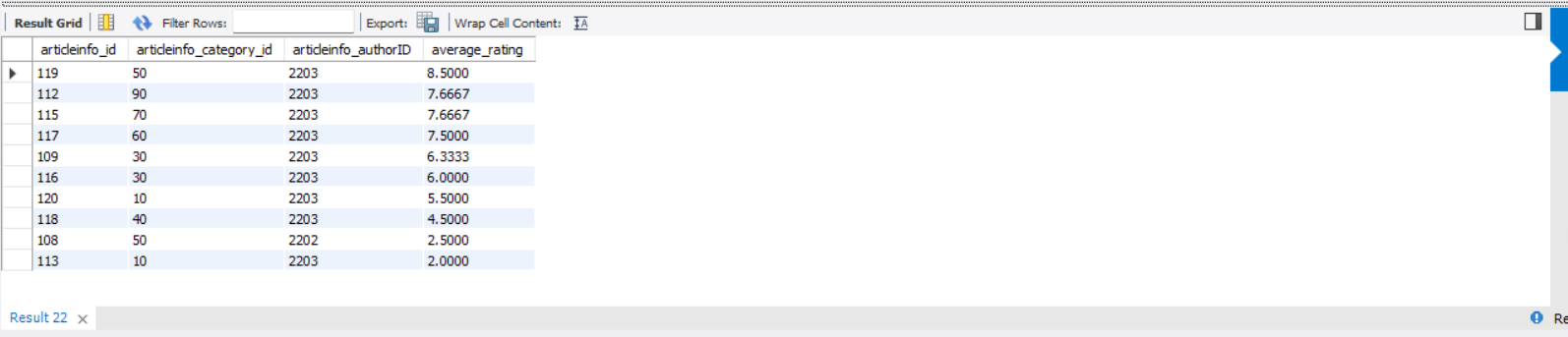
GROUP BY articleinfo\_id

ORDER BY average\_rating DESC

Algjebër relacionale:

π(articleinfo\_id, articleinfo\_category\_id, articleinfo\_authorID, AVG(article\_rating) as average\_rating) (σ(articleinfo\_id = articleratings\_id) (articleinformation ⨝ articleratings)) GROUP BY articleinfo\_id ORDER BY average\_rating DESC

Rezultati:



1. Për të zgjedhur cilët përdorues preferojnë një artikull specifik

SELECT userpref\_id, userpref\_username, articleinfo\_category\_id, articleinfo\_title

FROM userpreferences

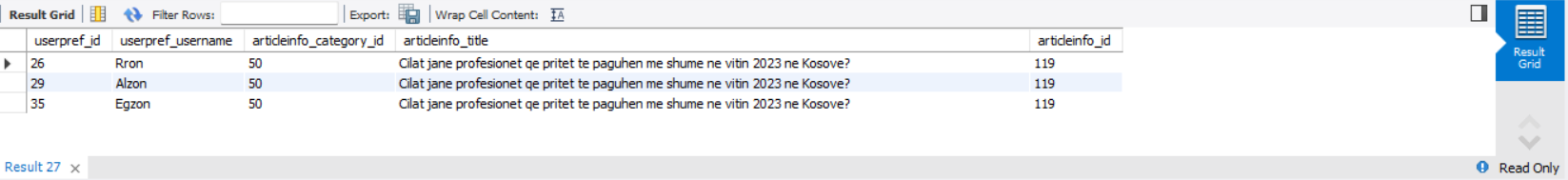
JOIN articleinformation ON userpref\_category\_id = articleinfo\_category\_id

WHERE articleinfo\_id = '119';

Algjebër relacionale:

π userpref\_id, userpref\_username, articleinfo\_category\_id, articleinfo\_title ( userpreferences ⋈\_{userpref\_category\_id = articleinfo\_category\_id} (π articleinfo\_category\_id, articleinfo\_title ( articleinformation ⋈\_{articleinfo\_id = '119'}) )

Rezultati:



1. Për të gjetur emrin e autorit i cili ka shkruar artikullin më të preferuar

SELECT author\_lname, author\_inital, author\_name, articleinfo\_id, articleinfo\_title

FROM articleinformation

JOIN author a ON articleinfo\_authorID = author\_id

JOIN userpreferences ON articleinfo\_category\_id = userpref\_category\_id

GROUP BY author\_lname, author\_inital, author\_name, articleinfo\_id, articleinfo\_title

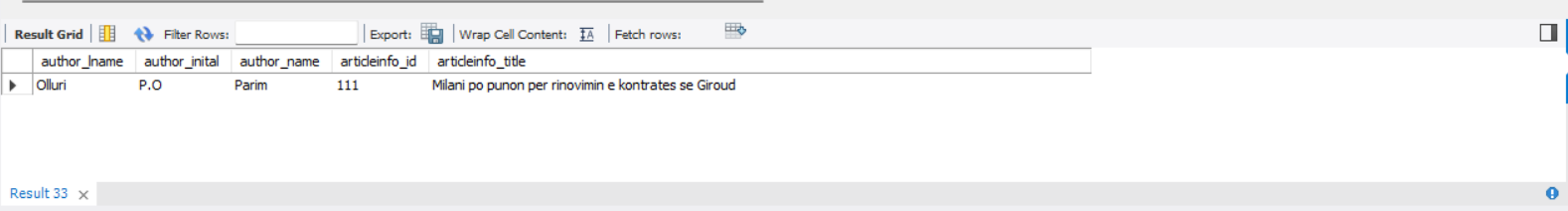
ORDER BY COUNT(\*) DESC

LIMIT 1;

Algjebër relacionale:

π author\_lname, author\_inital, author\_name, articleinfo\_id, articleinfo\_title ((Articleinformation⋈\_{articleinfo\_category\_id = user\_pref\_category\_id} userpreferences) ⟪article\_author⟫ COUNT(\*) ⟪article\_author⟫ DESC LIMIT 1 )

Rezultati:



1. Gjeni 3 kategoritë më të preferuara

SELECT category\_name

FROM Categories

LEFT JOIN userpreferences ON category\_id = userpref\_category\_id

GROUP BY category\_name

ORDER BY COUNT(userpref\_category\_id) DESC

LIMIT 3;

Algjebër relacionale:

π category\_name (

(Categories ⟨⋈\_{category\_id = userpref\_category\_id} userpreferences⟩)

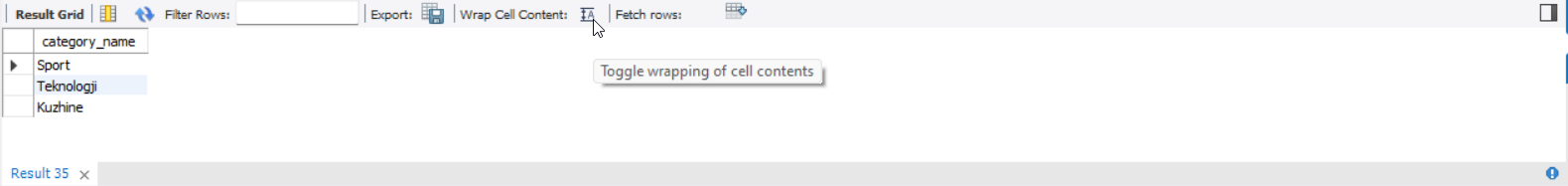
⟪category\_name⟫ COUNT(userpref\_category\_id)

⟪category\_name⟫ DESC

LIMIT 3

)

Rezultati:



1. Gjeni kategorinë më pak te preferuar

SELECT category\_name

FROM Categories

LEFT JOIN userpreferences ON category\_id = userpref\_category\_id

GROUP BY category\_name

ORDER BY COUNT(userpref\_category\_id) ASC

LIMIT 1;

Algjebër relacionale

π category\_name ( (Categories ⟨⋈\_{category\_id = user\_prefcategory\_id} userpreferences⟩) ⟪category\_name⟫ COUNT(userpref\_category\_id) ⟪category\_name⟫ ASC LIMIT 1 )

Rezultati:



1. Për të zgjedhur numrin e përdoruesve nga secili qytet:

SELECT user\_city, COUNT(\*) AS num\_users

FROM users

JOIN useraddress ON users.user\_zip = useraddress.user\_zip

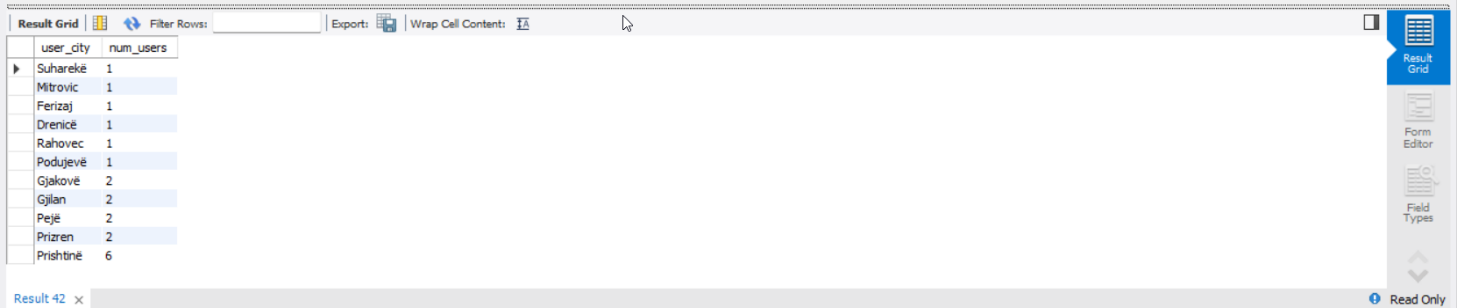
GROUP BY useraddress.user\_city

ORDER BY num\_users;

Algjebër relacionale:

π users.user\_city, COUNT(\*) AS num\_users ( (Users ⋈\_{users.user\_zip = useraddress.user\_zip} useraddresses) ⟪useraddress.user\_city⟫ )

Rezultati:



1. Për një përdorues specifik për të nxerrur cilat artikuj të i’a rekomandoni në bazë të preferencave të tyre në kategori të caktuara

SELECT articleinfo\_id, articleinfo\_title, articleinfo\_category\_id

FROM articleinformation

JOIN userpreferences ON articleinfo\_category\_id = userpref\_category\_id

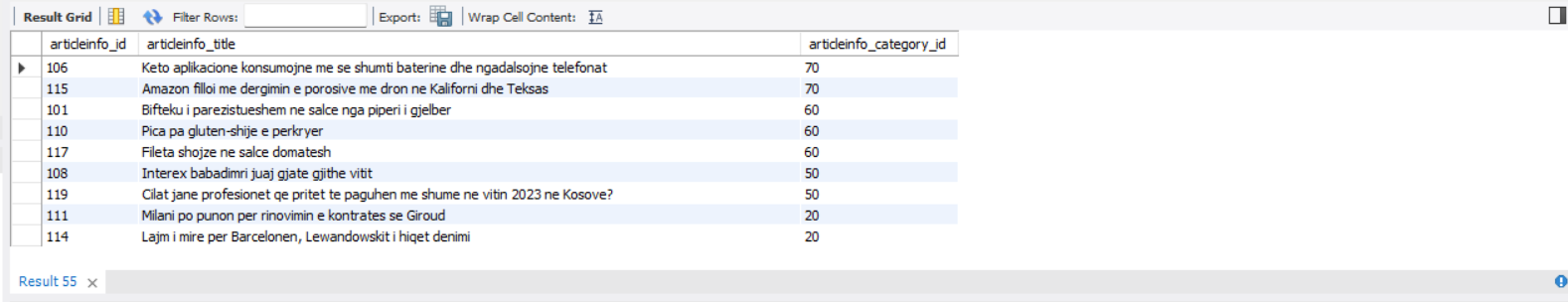
WHERE userpreferences.userpref\_username = 'Egzon'

ORDER BY userpref\_category\_id DESC;

Algjebër relacionale:

π articleinfo\_id, articleinfo\_title, articleinfo\_category\_id ( (articleinformation ⟨⋈\_{articleinfo\_category\_id = userpref\_category\_id} userpreferences⟩) ⟪userpreferences.userpref\_username = 'Egzon'⟫ ⟪userpref\_category\_id⟫ DESC )

Rezultati:



1. Për të nxjerrur vendin nga vijnë të gjithë user-atë cilët preferojnë një kategori të caktuar p.sh “Teknologji”.

SELECT users\_username, user\_city

FROM users

JOIN useraddress ON users.user\_zip = useraddress.user\_zip

JOIN userpreferences ON users\_username = userpref\_username

JOIN categories ON userpref\_category\_id = category\_id

WHERE category\_name = 'Teknologji';

Algjebër relacionale:

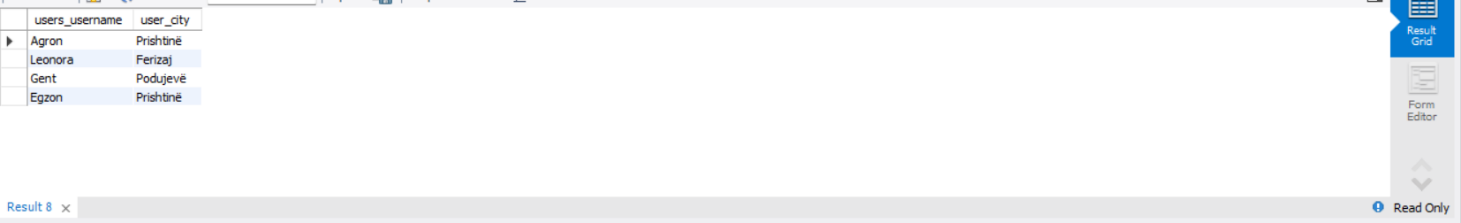
π users\_username, user\_city (

(users ⟨⋈\_{users.user\_zip = useraddress.user\_zip} useraddress⟩ ⟨⋈\_{users\_username = userpref\_username} userpreferences⟩ ⟨⋈\_{userpref\_category\_id = category\_id} categories⟩)

⟪category\_name = 'Teknologji'⟫

)

Rezultati:



1. Për të nxjerrur të gjitha komentet për një artikull të caktuar

SELECT \*

FROM commentcontent

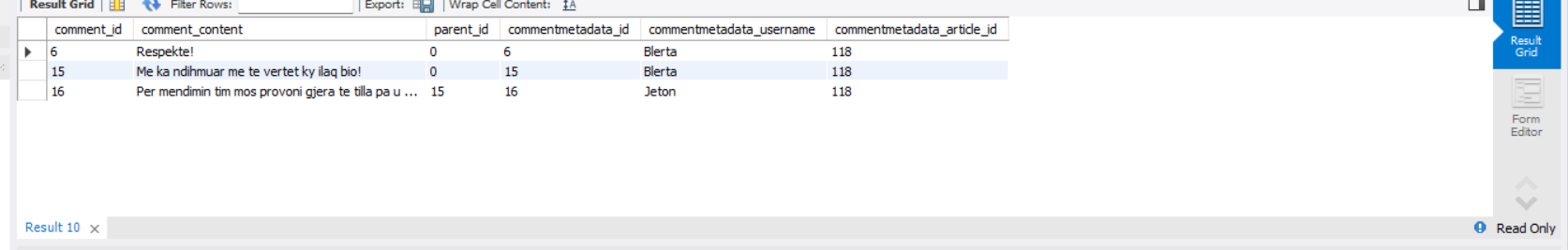
JOIN commentmetadata ON comment\_id = commentmetadata\_id

WHERE commentmetadata\_article\_id = '118';

Algjebër relacionale :

π \* ( (commentcontent ⟨⋈\_{comment\_id = commentmetadata\_id} commentmetadata⟩) ⟪commentmetadata\_article\_id = '118'⟫ )

Rezultati :



1. Për të nxjerrur artikujt më të komentuar :

SELECT articleinfo\_id,articleinfo\_title, articleinfo\_category\_id, commentmetadata\_username, COUNT(articleinfo\_id) AS comments

FROM articleinformation

JOIN commentmetadata ON articleinfo\_id = commentmetadata\_article\_id

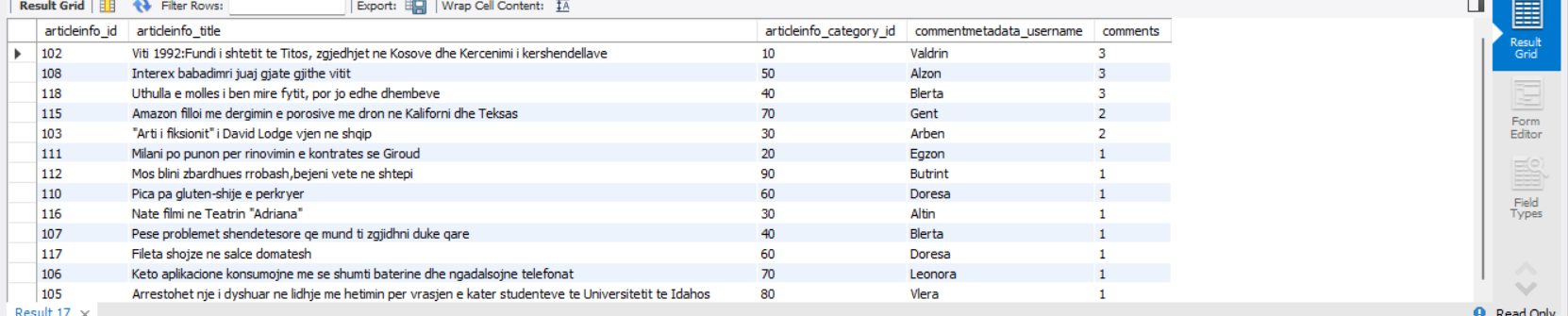
GROUP BY articleinfo\_id

ORDER BY comments DESC

Algjebër relacionale :

π(articleinfo\_id,articleinfo\_title, articleinfo\_category\_id, commentmetadata\_username, COUNT(articleinfo\_id) as comments) (σ(articleinfo\_id = commentmetadata\_article\_id) (articleinformation ⨝ commentmetadata)) GROUP BY articleinfo\_id ORDER BY comments DESC

Rezultati :



1. Për të nxjerrur përdoruesit që kanë komentuar në një artikull specifik :

SELECT DISTINCT users\_username, commentmetadata\_article\_id, articleinfo\_title

FROM users

JOIN commentmetadata ON users\_username = commentmetadata\_username

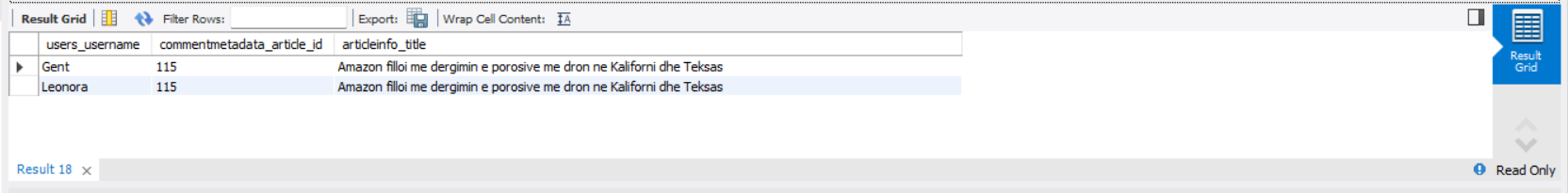
JOIN articleinformation ON commentmetadata\_article\_id = articleinfo\_id

WHERE articleinfo\_id = '115';

Algjebër relacionale:

π(DISTINCT users\_username, commentmetadata\_article\_id, articleinfo\_title) (σ(articleinfo\_id = '115') (users ⨝ commentmetadata ⨝ articleinformation))

Rezultati:



Veglat e perdorura :

1. MySQL Workbench - është një mjet vizual që shërben për të punuar me SQL servera dhe baza të të dhënave.
2. SmartDraw – është një faqe që shërben për krijimin e dokumenteve dhe dizajnimin e diagrameve të ndryshme. Është mjaft lehtë për t’u përdorur pasi që ka veglat e nevojshme dhe përmban diku mbi 70 shabllone.

Përdor nivelin më të lartë të enkriptimit për të ruajtur dhe transferuar të dhëna për të garantuar siguri.

Kontributi i anëtarëve të grupit :

Albina Grajqevci-ka kontribuar me dizajnimin e bazes se te dhenave,implementimin e databazes ne MySQL, krijimin e arkitekturës së projektit duke bërë diagramin e :Crow’s Foot, UML dhe Object-Oriented.

Diellza Bajraktari - ka kontribuar ne dizajnimin e bazes se te dhenave, implementimin e databazes ne MySQL, normalizmin e databazës, krijimin e query-ve dhe shprehjet e algjebres relacionale per query-t e dhene.

Dalina Elshani- ka kontribuar ne dizajnimin e bazes se te dhenave, implementimin e databazes ne MySQL.

Referencat

(1) <https://journals.sagepub.com/doi/full/10.1177/0165551519894928?journalCode=jisb>

(2) <https://journals.sagepub.com/doi/full/10.1177/0165551519894928?journalCode=jisb>

(3) <https://www.khanacademy.org/computing/computer-programming/sql>

(4) <https://www.smartdraw.com/about/>

(5) <https://telegrafi.com/lajme/>

(6) (6) <https://www.gazetaexpress.com/>

(7)<https://dl.ebooksworld.ir/motoman/Cengage.Learning.Database.Systems.13th.Edition.www.EBooksWorld.ir.pdf>