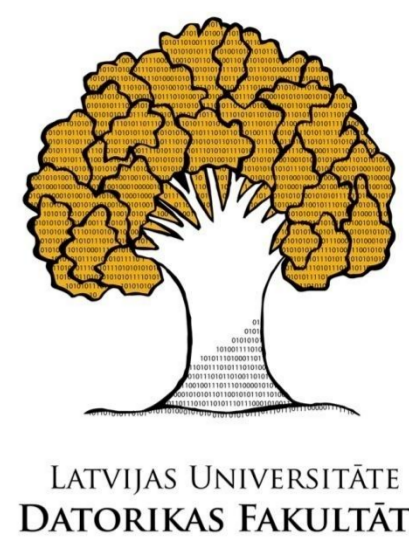


Modeļu bāzēto arhitektūru attīstība



G. Arnicāns, J. Bičevskis, Ģ. Karnītis
V. Arnicāne, I. Medvedis

DF | Datorikas fakultāte



Levads

Daudzas mūsdienu informatīvās sistēmas datus glabā relāciju datubāzēs.

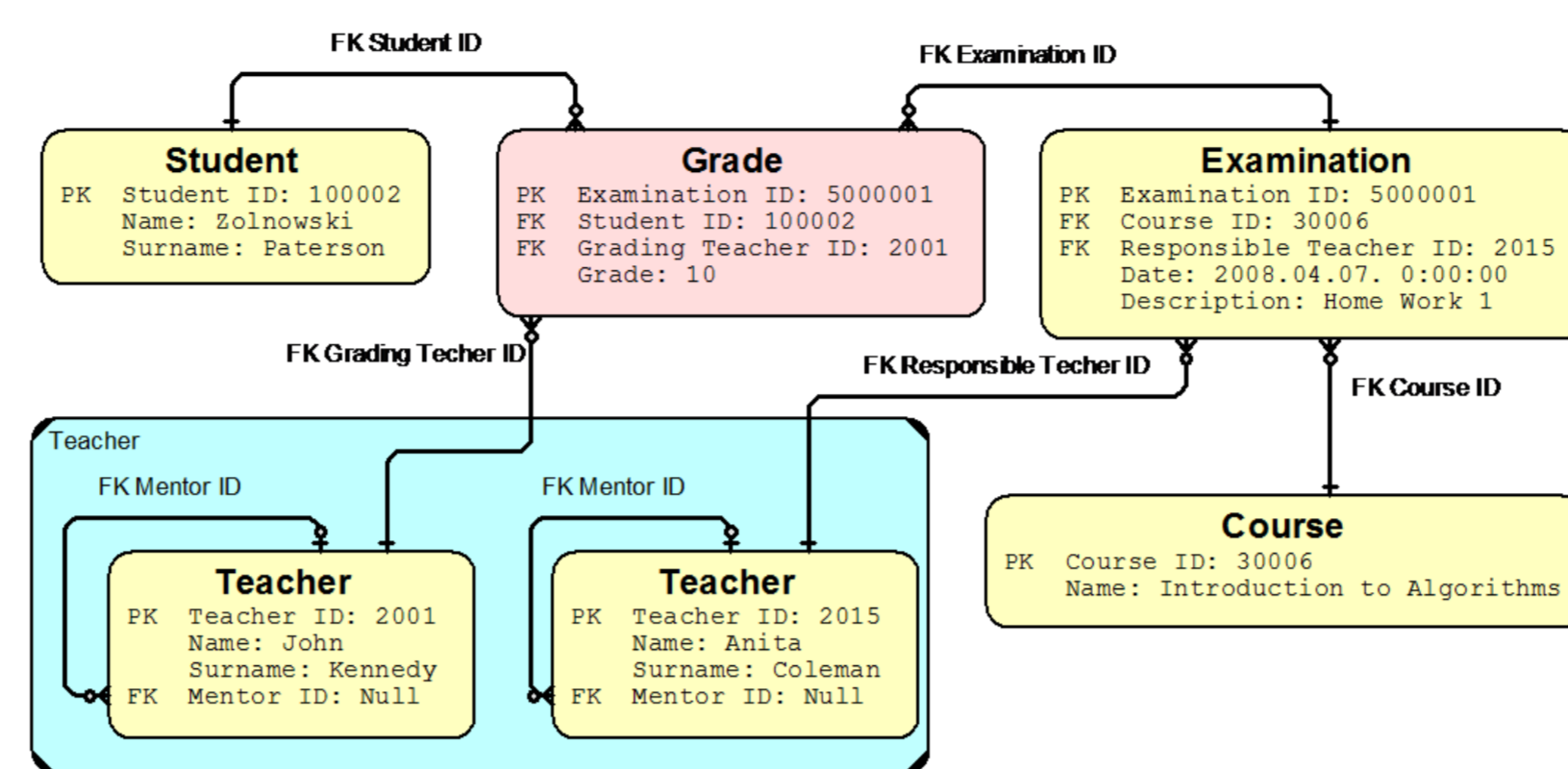
Datu ieguvei no datubāzes pamatā tiek izmantoti SQL valodā rakstīti pieprasījumi, kuru pierakstam nepieciešamas programmēšanas iemaņas.

SQL pieprasījumi var saturēt kļūdas un, tos izpildot, lietotājs var saņemt nekorektus datus.

Alternatīvs veids datu ieguvei un datu aplūkošanai ir datubāzes pārvaldības sistēmu izmantošana, kas parasti piedāvā vienas tabulas pārliūkošanu.

Diemžēl lietotājs redz ierobežotu datu kopu un nepieciešamas specifiskas IT zināšanas saistīto datu ieguvei.

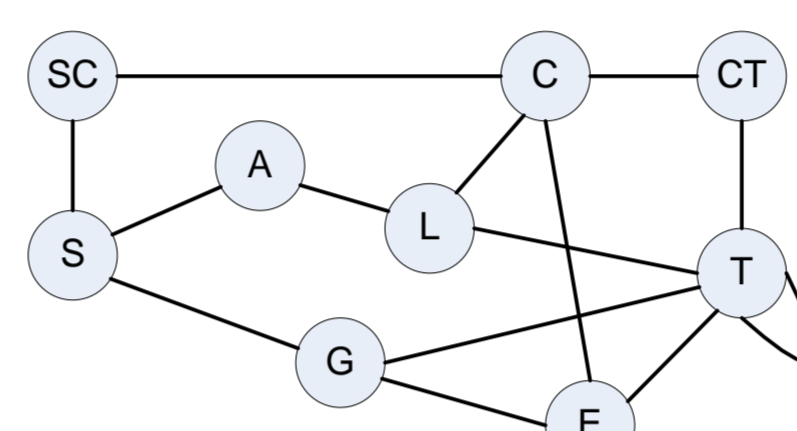
Ir izstrādāta metodoloģija, kas nodrošina datubāzes datu pārliūkošanu bez programmēšanas prasmēm un kas lietotājam piedāvā iespēju skatīt savstarpēji integrētus datus.



Attēls 3. Fragments no datu grafa. Centrālais ir ieraksts par atzīmi (Grade). Ar relācijām saistītie ieraksti sniedz papildus informāciju par šo atzīmi.

Augstākā abstrakcijas līmenī datu grafu apraksta datubāzes metadati. Tabulas un relācijas veido grafu (Attēls 4)

Attēls 4. Datubāzes tabulas un relācijas veido grafu. Piemērs atbilst diagrammai Attēlā 1. Virsotņu burti atbilst tabulu nosaukumu sākuma burtiem



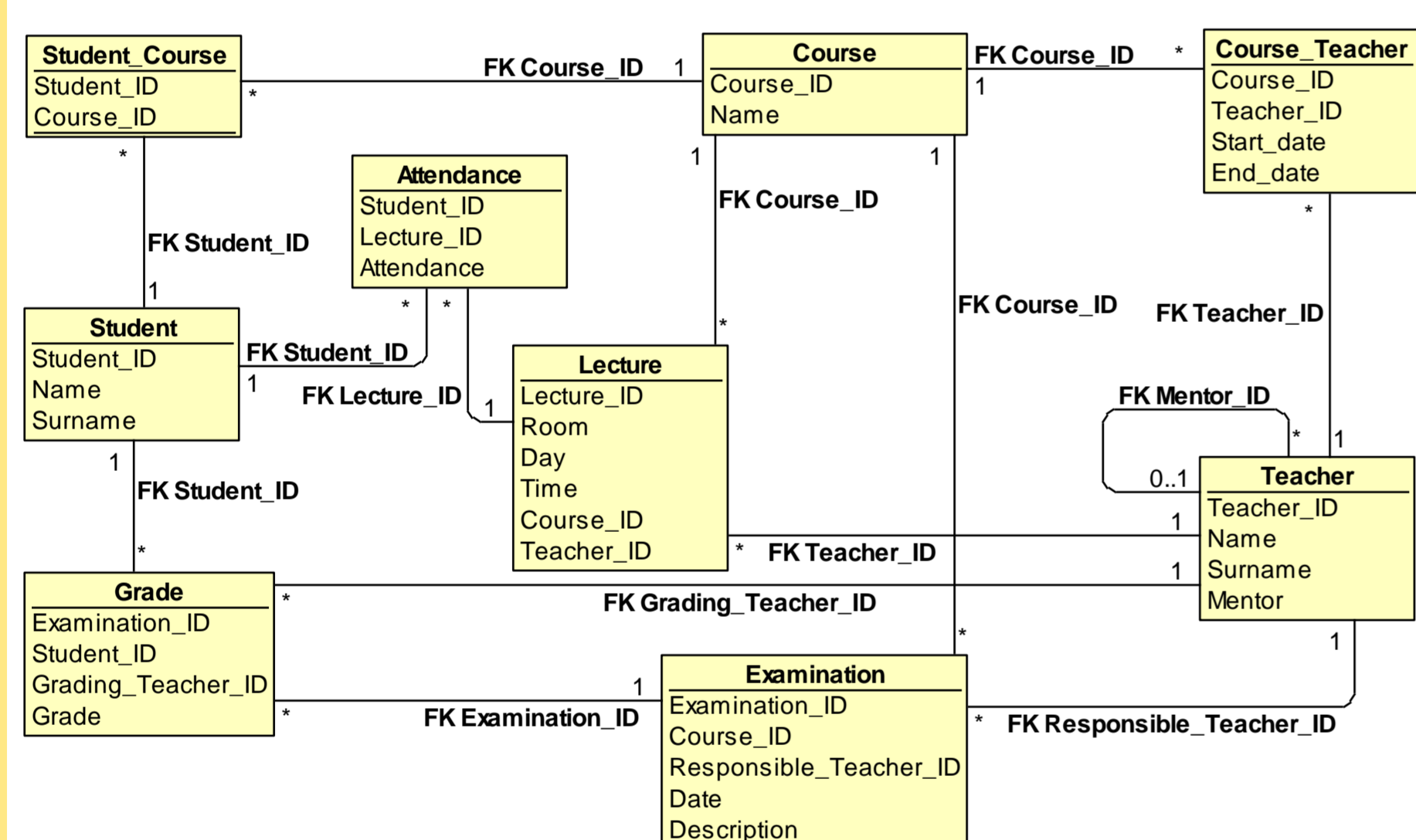
Katra šķautne tiek aizstāta ar 2 orientētām relācijām (bultām) un tiek iegūts datu pārliūkošanas grafs (Attēls 5).

Ir atrastas vairākas stratēģijas, kas, ņemot vērā grafa topoloģiju un bultu īpašības, spēj atlasīt konkrētam ierakstam saistīto informāciju no daudzām tabulām. Būtiski ir tas, ka nav nepieciešams zināt tabulu semantiku. Datu atlasē piemērs atzīmes ierakstam, pielietojot predefinētu atlasī, dots Attēlā 3.

Attēls 5. Datu pārliūkošanas grafs. Katra bulta nosaka sagaidāmo ierakstu daudzumu saistītajās tabulās

Problēma

Informācija relāciju datubāzēs tiek glabāta daudzās datu tabulās, starp kurām ir definētas attiecības (Attēls 1). Lai izgūtu datus no datubāzes, programmētājam jāpārzina datu modelis un jāprot rakstīt pieprasījumus ar speciālas programmēšanas valodas palīdzību, piemēram, SQL.



Attēls 1. Piemērs nelielai relāciju datubāzei, kas satur datus par studijām universitātē. Taisnstūris reprezentē datu tabulu ar tās nosaukumu un datu laukiem. Līnijas starp taisnstūriem reprezentē attiecības (relācijas) starp tabulām, t.i. starp to ierakstiem. Piemēram, katrai lekcijai (Lecture) ir zināms tās mācību priekšmets (Course) un katram mācību priekšmetam var būt daudzas lekcijas.

Datubāzu pārvaldības rīki parasti nodrošina datu apskati tabulas veidā (Attēls 2). Dati parādās kodētā veidā atbilstoši tabulas definīcijai, skats uz datiem "ierobežots".

ExamID	StudentID	Grade	TeacherID
5000001	100001	2001	5
5000001	100002	2001	10
5000001	100003	2001	0

Attēls 2. Šādi datus redz programmētājs tabulām Grade un Teacher, izmantojot tradicionālu rīku datubāzes satura aplūkošanai.

Lai redzētu saistītos datus, programmētājam ir jāraksta nelielas programmas. Tas prasa zināšanas, pūles un potenciāli var uzrakstīt kļūdainus pieprasījumus.

Risinājums

Datu ieraksti tabulās, ņemot vērā relācijas starp tabulām, veido lielu datu grafu – virsotnes ir tabulu ieraksti, šķautnes ir relācijas starp ierakstiem (Attēls 3).

Datu pārliūkošanu par definēt kā pārvietošanos datu grafā no vienas virsotnes (ieraksta) uz citu. Katrā brīdī tiek attēlota ne tikai virsotne, kurā atrodamies, bet arī ar to saistītie dati, kas atlasīti ar kādu predefinētu stratēģiju.

Rezultāti

Ir izveidots datubāzes pārliūks **DigiBrowser**, kas darbojas ar *Oracle*, *MySQL*, *Postgress* un *MS SQL* datubāzēm. Izveidotais pārliūks eksperimentālā režīmā ir ieviests vairākos projektos. Pašlaik notiek pētījumi, lai piemērotu izstrādāto tehnoloģiju RDF datubāzu pārliūkošanai.

Bez **DigiBrowser** rīka ir izveidoti vairāki citi pārliūka prototipi, lai atrastu labākās datu atlasē un integrācijas stratēģijas, kā arī lai pārbaudītu dažādas datu saskarnes.

Startējot **DigiBrowser** lietotājs izvēlas datubāzi, kuru vēlas pārliūkot (piemēram, attēloto Attēlā 1). Pēc pieslēgšanās ekrāna kreisajā pusē redzamas visas tabulas (Attēls 7). Izvēlamies tabulu *Teacher*. Rīks parāda visus docētājus saprotamā veidā, izmantojot predefinēto stratēģiju (labā augšējā daļa). Var redzēt arī konkrēti izvēlētam docētājam saistīto informāciju (labā apakšējā daļa).

Attēls 7. Docētāja "John Kennedy" dati un visas viņa lasītās lekcijas.

Uzklīkšķinot ar peli uz cilnes *Grade*, iegūstam konkrētā docētāja visas piešķirtās atzīmes (Attēls 8).

Attēls 8. Docētāja "John Kennedy" dati un visas viņa piešķirtās atzīmes.

Lai veiktu pārvietošanos pa datubāzi, izmantojam saites līdzīgi kā tīmeklī. Var "aizceļot" gan uz vienu konkrētu objektu, gan objektu kopu. Piemēram, ar dubultklikšķi uz cilnes *Grade* par centrālajiem objektiem kļūst atzīmes (labā augšējā daļa Attēlā 9), kas bija pieejamas (labā apakšējā daļa Attēlā 8).

Attēls 9. Docētāja "John Kennedy" piešķirtās atzīmes un detalizēta informācija par šiem vērtējumiem.

Analoģiskas informācijas iegūšanai (tikai labā augšējā daļa) programmētājam būtu jāraksta

```

Select top 100
g.Examination_ID, g.Student_ID, g.Grading_teacher_ID, g.Grade,
(Select top 1 t.Description from Examination t
 where t.Examination_ID=g.Examination_ID) as xExamination_ID,
(Select top 1 coalesce(cast("Name" as varchar(250)), '') + ' ' +
coalesce(cast("Surname" as varchar(250)), '')) as xFormula_
from Teacher t
 where t.Teacher_ID=g.Grading_teacher_ID) as xGrading_teacher_ID,
(Select top 1 coalesce(cast("Name" as varchar(250)), '') + ' ' +
coalesce(cast("Surname" as varchar(250)), '')) as xFormula_
from Student s
 where s.Student_ID=g.Student_ID) as xStudent_ID
from Grade g
 where (g.Grading_teacher_ID in (2001))
    
```

Noslēgums

Apskatītā datubāzes pārliūkošanas tehnoloģija un izstrādātie rīki eksperimentālajos lietojumos ir uzrādījuši labus rezultātus. Pārliūki var palīdzēt gan informatīvo sistēmu izstrādē, gan to testēšanā un uzturēšanā, gan arī vienkāršai datu analīzei.

Tiek turpināts darbs pie jaunu datu saistīšanas un attēlošanas stratēģiju izstrādes, kā arī lietotāja saskarnes uzlabošanas.

Kontakti

Latvian University
Datorikas fakultāte

Adrese: Rīga, Raiņa bulv. 19 – 332
Telefons: +371 67034475
E-pasts: Guntis.Arnicans@lu.lv

Pateicības

Darbs veikts ar ESF atbalstu projektā

„Datorzinātnes pielietojumi un tās saiknes ar kvantu fiziku”
Nr. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044



IEGULDĪJUMS TAVĀ NĀKOTNĒ

