

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

5.praktiskais darbs

**XML un JASON datu bāzes, to īpašības un iespējas**

Izstrādāja: Natans Šalamberidze

171RMC203

Pārbaudīja: lektors Jānis Eiduks

2023./2024. māc. Gads

**Darba uzdevums**

1. Jāizprojektē divas tabulas ar XML dokumentu glabāšanu.

2. Ar izvēlētu XML redaktoru jāizveido izprojektētie XML dokumentu faili (grafiskais attēlojums jāiekļauj darbā). Tiem jānoģenerē atbilstošie shēmu faili.

3. Binārās glabāšanas XMLType tipa tabulu ar shēmas izmantošanu definēšana. Tagad šis XML datu glabāšanas veids ir galvenais (sākot ar Oracle 11.2 versiju).

4. Datu ievade izveidotajās tabulās (varianti: tekstveida, lielo objektu BLOB un SQL\*Loader izmantošana).

5. XMLTable vaicājumu izpilde (2 vaicājumi).

6. XMLQuery vaicājumu izpilde (2 vaicājumi).

7. XMLCast vaicējumu izpilde (2 vaicājumi)

8. XML datu izvadīšana relāciju datu veidā (XML dokumenta datu transformēšana relāciju datos).

9. Relāciju datu izvadīšana XML datu formātā (relāciju datu transformēšana XML datu formātā).

10. XML dokumentus pārrakstīt JSON valodā. Izmantot kādu JSON redaktoru.

11. Tabulu izveidošana JSON datu glabāšanai (tabulas definēšana, datu ievade).

12. Datu izgūšana no tabulas ar JSON datu glabāšanu. Izmantot analoģiskas funkcijas XML izgūšanai.

13. XML un JSON valodas izmantošanas datu bāzē salīdzinājums.

14. Secinājumi (kas labs, kas slikts, kas noskaidrots, XML un JASON izmantošanas iespēju izvērtējums).

1-3) Sākuma tika veidoti divi XML faili programā Oxygen XML Editor 26.0

A screenshot of a computer

Description automatically generatedArī tika ģenerētas shēmas abiem failiem.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Pēc tam mapē kur tiek glabāti XML un XSD faili tika reģistrēta kā direktorijā.

A screenshot of a computer

Description automatically generated

XML shēmas tika reģistrēti.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedPārbaude:

Pēc tam tika izveidotas divas binārās glabāšanas XMLType tipa tabulas ar shēmas izmantošanu.

A screenshot of a computer program

Description automatically generated

Pārbaude:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated4) Lai ievietot XML failus tabulās tika izveidota procedūra:

A screenshot of a computer

Description automatically generatedProcedūras palaišana:

A screenshot of a computer

Description automatically generatedPārbaude:

A screenshot of a computer

Description automatically generated

5) No XML failā, kas saturēja informāciju par grāmatām ar XMLTable vaicājuma palīdzību tika izgūta informācija par grāmatām, kas ir latviešu valodā un kuram vērtējums ir lielāk par 4.5

A screenshot of a computer

Description automatically generated

No XML failā, kas saturēja informāciju par uzņēmuma projektiem ar XMLTable vaicājuma palīdzību tika izgūta informācija par projektu ilgumu gados un mēnešos.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated6) No XML failā, kas saturēja informāciju par grāmatām ar XMLQuery vaicājuma palīdzību tika izgūta informācija par grāmatu izdošanas gadu un vērtējumiem.

No XML failā, kas saturēja informāciju par uzņēmuma projektiem ar XMLQuery vaicājuma palīdzību tika izgūta informācija par darbiniekiem, kas piedalījās jebkurā projektā.

A screenshot of a computer

Description automatically generated

7) No XML failā, kas saturēja informāciju par grāmatām ar XMLCast vaicājuma palīdzību tika izgūta informācija par grāmatu vidējo vērtējumu.

A screenshot of a computer

Description automatically generated

No XML failā, kas saturēja informāciju par uzņēmuma projektiem ar XMLCast vaicājuma palīdzību tika izgūta informācija par visiem projekta vadītajiem.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated8) XML datu par uzņēmuma projektu izvadīšana relāciju datu veidā.

9. Relāciju datu izvadīšana XML datu formātā (relāciju datu transformēšana XML datu formātā).

Tabulas, kas saturēs relāciju datus par uzņēmuma projektiem, veidošana.

A screenshot of a computer

Description automatically generated

Datu ievade tabulā.

A white screen with a white border

Description automatically generated with medium confidence

XML datu iegūšana

A screenshot of a computer program

Description automatically generated

10. XML dokumentus pārrakstīšana JSON valodā ar rīka FreeFormater palīdzību.

A screenshot of a computer

Description automatically generated

11. Tabulu izveidošana JSON datu glabāšanai (tabulas definēšana, datu ievade).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

12. Datu izgūšana no tabulas ar JSON datu glabāšanu.

JSON\_TABLE izmantošana

A screenshot of a computer

Description automatically generated

JSON\_QUERY izmantošana

A white screen with a white background

Description automatically generated with medium confidence

JSON\_VALUE izmantošana

A screenshot of a computer

Description automatically generated

13-14. XML un JSON valodas izmantošanas datu bāzē salīdzinājums un Secinājumi (kas labs, kas slikts, kas noskaidrots, XML un JASON izmantošanas iespēju izvērtējums).

XML un JSON datu glabāšana Oracle datu bāzē ir ļoti līdzīga: pastāv atsevišķi datu tipi, vai arī var glabāt tādus datus arī kā tekstu. Pastāv līdzīgs mehānisms, lai atļaut glabāt tikai tādus XML vai JSON dokumentus, kas atbilst noteiktam izskatam (shēmai).

XMLTable un JSONTable funkcijas darbojas ļoti līdzīgi un ļauj ērti izmantot JSON vai XML datus kā relāciju datus. Savukārt JSONQuery funkcijai ir mazāk pielietojumu. Autoram neizdevās apvienot divus atribūtus no JSON failā vienā rezultātu kolonā, bet XML datiem tas izdevās diezgan viegli.

JSONValue un XMLCast ar XMLQuery funkcijas izpilda aptuveni vienu un to pašu funkciju, kas ļauj iegūt vienu atsevišķu vērtību.

**SECINĀJUMI**

Reģistrēt shēmas nebija viegli, bija jāpavada laiks, lai saprastu, kas tieši izraisa kļūdu, kad tika veikta XML faila ielāde. Tas sagādāja visvairāk grūtību. Pēc datu ielādes, vaicāt XML datus bija diezgan viegli, subjektīvi visērtāk bija izmantot XMLTable funkciju un strādāt ar XML datiem kā ar relāciju datiem. Iespējams, izmantot XMLQuery būtu ātrāk un efektīvāk pie lieliem datu apjomiem, bet ar mazo datu daudzumu nekādas neērtības netika pamanītas.

Dots praktiskais darbs ļāva saprast, ka JSON un XML datus var viegli integrēt, piemēram, kādā datu krātuvē, un pārvērst tos relāciju datos, lai veiktu kādu analītiku. Tas būtu ļoti noderīgi, jo mūsdienās daudz atbilžu no tīmekļa servisiem vai API tiek piedāvātas JSON vai XML formātā.