

Datu atlase

Datu atlase

- Nodrošina ar loģisko izteiksmju palīdzību
- Vairumā DBMS tas tiek veikts ar SQL valodas komandas SELECT palīdzību
- SQL (Structured Query Language) – universāla standartizēta valoda operācijām ar datu bāzēm

Loģiskās izteiksmes

Visās datu atlasēs komandās tiek izmatotas loģiskas izteiksmes.

Loģiskajai izteiksmei var būt divas vērtības

patiesa **TRUE**

nepatiesa **FALSE**

Loģiskās izteiksmes

Loģiskā izteiksme ir **mainīgā** salīdzinājums ar citu **mainīgo** vai **konstanti**

Loģiskās izteiksmes rezultāts ir viena no divām loģiskām vērtībām – **True** (patiess) vai **False** (nepatiess)

Ja konstatēta atbilstība, izteiksmes vērtība ir **TRUE**, ja nē, tad – **FALSE**

Loģiskās izteiksmes

- Loģiskajā izteiksmē kā **mainīgo** var izmantot datu tabulas **lauku**
- Tad ar konstanti vai citu mainīgo tiek salīdzinātas katram tabulas **ierakstam** atbilstošās izvēlētā lauka vērtības
- Ja konstatēta atbilstība, izteiksmes vērtība, konkrētajam ierakstam **TRUE**, ja nē, tad – **FALSE**
- Tiek atlasīti tie ieraksti, kuriem izteiksmes vērtība ir **patiesa (TRUE)**.

Tabula

Pieņemsim, ka mūsu rīcībā ir tabula ar Latvijas putnu sugu sarakstu, kur ir lauki

Suga, gints sugas, ģints nosaukums

Dzimta dzimtas nosaukums

ES loģiskais lauks, piederība ES aizsargājamo sugu sarakstam

Olu_skaitis vidējais olu skaits ligzdā

Loģiskās izteiksmes

Izteiksmēs parasti norāda lauku nosaukumus.

Piemērs.

Lauks “ES” ir loģiskais lauks, kurā norādīts vai suga ir Eiropā aizsargājama.

Šim laukam var būt tikai divas vērtības (**TRUE** vai **FALSE**).

Ja vēlas atlasīt Eiropā aizsargājamās sugas, tad loģiskā izteiksme ir ļoti vienkārša:

ES

Operators NOT

Loģiskajās izteiksmēs var izmatot loģiskos operatorus.

Operators **NOT** apgriež izteiksmes vērtību.

Izteiksme

NOT *ES*

būs **patiesa**, ja suga nebūs Eiropā aizsargājama.

Vērtību salīdzināšana

Ja loģiskajā izteiksmē iekļaujama lauks vai mainīgais nav loģiskais, tad tas ir jāsalīdzina ar kādu konstanti, izmantojot operatorus:

=

< vai >

<= vai >=

Piemērs

Izteiksme

dzimta = “Passeridae”

būs patiesa visām putnu sugām, kas pieder zvirbuļu dzimtai.

Izteiksme

olu_skaits ≥ 4

būs patiesa visos gadījumos, kad olu skaits ir lielāks vai vienāds par 4.

Operatora NOT izmantošana

Izteiksme

NOT dzimta = “Passeridae”

būs patiesa visām putnu sugām, kas **nepieder** zvirbuļu dzimtai.

Izteiksme

NOT olu_skaits ≥ 4

būs patiesa visos gadījumos, kad olu skaits ir mazāks par 4.

Izteiksme ar vairākiem laukiem

Ja atlasi grib veikt pēc vairākiem parametriem, loģiskās izteiksmes daļas apvieno, izmantojot loģiskos operatorus

AND

OR

Operators AND

Loģiskās izteiksmes daļas apvienojot ar operatoru AND, izteiksme būs patiesa tikai tad, ja abas apvienojamās daļas būs patiesas. **Izteiksme**

`dzimta = "Passeridae" AND olu_skaitis >= 4`

ir patiesa, ja suga pieder zvirbuļu dzimtai un olu skaits ligzdā ir lielāks vai vienāds ar 4.

Piemērs

Izteiksme

dzimta = “Passeridae” AND dzimta = “Paridae”

nebūs patiesa **nevienā** gadījumā, jo vienā ierakstā vienā laukā nevar būt divas dažādas vērtības

Operators OR

Loģiskās izteiksmes daļas apvienojot ar operatoru OR, izteiksme būs patiesa tad, ja kaut viena no apvienojamām daļām būs patiesa. **Izteiksme**

dzimta = “Passeridae” OR dzimta = “Paridae”
ir patiesa, ja suga pieder zvirbuļu vai zīlīšu dzimtai.

Piemērs

Izteiksme

dzimta = “Passeridae” OR olu_skaits \geq 4

ir patiesa **visām** zvirbuļu dzimtas sugām, **kā**
arī citu dzimtu sugām, kurām ligzdā
konstatētas vismaz 4 olas.

Operatoru izpildes kārtība

Līdzīgi kā aritmētisko operatoru gadījumā arī loģiskie operatori izpildās noteiktā secībā:

NOT

AND

OR

Piemērs

Izteiksme

$\text{NOT dzimta} = \text{“Passeridae” AND olu_skaits} \geq 4$
būs patiesa, ja olu skaits lielāks vai vienāds
par 4 **un** dzimta **nav** “Passeridae”.

Izteiksme

$\text{dzimta} = \text{“Passeridae” AND NOT olu_skaits} \geq 4$
būs patiesa, ja olu skaits **mazāks** par 4 **un**
dzimta ir “Passeridae”.

Piemērs

Izteiksme

dzimta = “Passeridae” OR dzimta = “Paridae”
AND olu_skaits => 4

ir patiesa **visiem** dzimtas “Passeridae”
pārstāvjiem, kā arī **tiem** dzimtas “Paridae”
pārstāvjiem, **kam** olu skaits ligzdā lielāks
vai vienāds ar 4.

Piemērs

Izteiksme

(dzimta = “Passeridae” OR dzimta = “Paridae”)
AND olu_skaits => 4

ir patiesa **visām** “Passeridae” un “Paridae”
dzimtu sugām, **ja** olu skaits lielāks vai
vienāds ar 4.

Piemērs

Izteiksme

NOT dzimta = “Passeridae” OR dzimta = “Paridae”
ir patiesa visām dzimtām, kas **nav** “Passeridae”

Izteiksme

NOT (dzimta = “Passeridae” OR dzimta = “Paridae”)
ir patiesa visām dzimtām, kas **nav ne**
“Passeridae”, **ne** “Paridae”

SQL valoda

SQL – Structured Query Language

Universāla valoda, kas ļauj veikt dažādas darbības ar datu bāzēm.

Visbiežāk izmanto SQL valodas komandu **Select**, kas domāta datu atlasei.

SQL Access vidē

- Datu atlasei tiek veidoti datu atlases formas
– *Queries*
- Veido izvēlnē *Create – Query Design*
- Tālāk ir jāpievieno tabula

Query Designer

- Select komandu var veidot ar *Query Designer* palīdzību, norādot parādāmos laukus, kārtotāšanas secību un atlasē komandas
- Izvēlnē *View* pārslēgties uz *SQL View*, kur redzama **Select** komanda vai uz *Datasheet View*, kur redzami atlasītie dati

Select komandas minimums

Select *

From *tabulas_nosaukums*

Atlasa visus laukus un visus ierakstus no norādītās tabulas.

Select komandas sintakse

Select *lauku nosaukumi*

From *tabulas nosaukums*

Where *izteiksme datu atlasei*

Order By *lauki pēc kuriem kārtot*

Select komandas minimums

Select *

From dzivnieki

Atlasa visus laukus un visus ierakstus no
tabulas *dzivnieki*.

Lauku izvēle

**Select gints, suga, autors, lsg, ASS ;
From dzivnieki**

**Parāda piecus norādītos laukus un visus
ierakstus.**

Lauku pārsaukšana

Laukus var pārsaukt, izmantojot komandu **As**, piemēram,

```
Select gints As gints_latiniski,  
      suga As suga_latiniski,  
      autors, lsg  
From dzivnieki
```

Tiek pārsaukti lauki *gints* un *suga*.

Datu kārtošana

Izmantojot komandu **Order by**, piemēram,
Select gints **As** gints_latiniski,
suga **As** suga_latiniski,
autors, lsg
From dzivnieki
Order by gints, suga

Datu kārtošana

Nosaukumu vietā var izmantot lauku
kārtas numurus

```
Select gints As gints_latiniski,  
      suga As suga_latiniski,  
      autors, lsg
```

```
From dzivnieki
```

```
Order by 1, 2
```

Datu atlase

Datu atlasei izmanto komandu Where, aiz kuras norāda datu atlases izteiksmi

Select gints, suga, autors, lsg, ASS ;

From dzivnieki

Where LSG = "1"

Datu atlase

Datu atlasei izmanto komandu Where, aiz kuras norāda datu atlases izteiksmi

Select gints, suga, autors, lsg, ASS ;

From dzivnieki

Where LSG = "1" or LSG="1"

Uzdevums

Izpildīt 4. uzdevumu –

\\priede\grozs\datu_bazes\uzdevums_4.doc