Duomenų bazių naudojimas, SQL

Jvadas j SQL kalbą, DBVS

Įvadas

37805203245

47005203214

46605203214

SELECT * FROM DARBUOTOJAI:

Duomenų bazė (toliau DB) - tai duomenų rinkinys, kurie organizuoti taip, kad būtų lengva juos peržiūrėti, tvarkyti ir atnaujinti.

600.00

300.00

800.00

PARDUOTUVES ID

3

4

·						
ASMENS_KODAS	VARDAS	PAVARDE	DIRBA_NUO	ATLYGINIMAS		
38705203214	ALGIS	GRIGAS	2012-10-10	600.00		
39705203214	ANTANAS	PALIONIS	2013-12-10	500.00		
38905212217	KAROLIS	KARALIUS	2015-10-10	1100.00		

GRIGAS

MARINA

GRIGAITIS 2012-10-10

2013-10-10

GARINA

VIKTORIJA PETKEVIC 1999-10-10

Operacijos ir sąvokos

DB vartotojų operacijos:

- 1. Naujo įrašo įvedimas (INSERT);
- 2. Esamo įrašo keitimas (UPDATE);
- 3. Esamo įrašo pašalinimas (DELETE);
- 4. Reikalingų įrašų paieška (SELECT).

Kompiuterinė sistema (darbo stotis, kompiuteris), kurioje atliekamos šios operacijos, vadinama duomenų bazių sistema (DBS).

Programinė įranga, suteikianti vartotojui galimybę dirbti su DB - duomenų bazių valdymo sistema (toliau - DBVS)

DBVS

Pagrindinė DBVS paskirtis - leisti vartotojui pažvelgti į DB kaip į aukštesnio lygio objektą, negu įrašų byla.

DBVS atlieka vartotojo sąsajos (interfeiso) su DB vaidmenį.

Vartotojas valdo DB pasitelkdamas formalią kalbą, pvz. SQL (*Structured Query Language* - struktūrizuotų užklausų kalba).

Pasitelkiant DBVS su DB galima bendrauti tiek tiesiogiai vartotojui, tiek programinei įrangai.

DBVS modeliai

Modelis - sąvokų rinkinys ir jų vartojimo taisyklės. DBVS modeliai:

- 1. Sąryšinis (reliacinis);
- 2. Hierarchinis;
- 3. Tinklinis.

Populiariausias - saryšinis. (1970 m., E. F. Kodas (E. F. Codd), IBM).

DB, apibrėžiamos naudojant reliacinį modelį, vadinamos reliacinėmis DB (RDB).

Atitinkamos jų valdymo sistemos - RDB valdymo sistemomis (RDBVS).

RDBVS sąvokos

Pagrindinės reliacinio modelio sąvokos:

- 1. Lentelė (table) aprašomi duomenys;
- 2. Stulpelis (column) aprašoma objekto charakteristika;
- 3. Eilutė (row) aprašomas vienas objektas;
- 4. Laukas/ reikšmė (value) eilutės ir stulpelio sankirta.

Reliacinės sistemos pagrindiniai principai:

- 5. Duomenys vartotojui pateikiami lentelėmis ir niekaip kitaip;
- 6. Vartotojui pateikiami operatoriai, leidžiantys iš esamų lentelių gauti naujas.

RDBVS pasirinkimai

Labiausiai paplitusios RDBVS:

- IBM DBVS DB2
- Oracle DBVS ORACLE
- Microsoft DBVS SQL Server
- Sybase Inc. DBVS SYBASE
- PostgreSQL atvirojo kodo
- MySQL atvirojo kodo

SQL

SQL - struktūrizuota užklausų kalba, naudojama reliacinėse DB sistemose

- 1. SQL yra deklaratyvi (neprocedūrinė) programavimo kalba.
- 2. SQL kalboje skiriamos trys sakinių grupės:
 - a. Duomenų apibrėžimo sakiniai (Data Definition Language DDL);
 - b. Duomenų manipuliavimo sakiniai (Data Manipulation Language DML);
 - c. Duomenų valdymo sakiniai (Data Control Language DCL).

$$SQL = DDL + DML + DCL$$

SQL = DDL + DML + DCL

DDL sakiniais kuriamos DB ir lentelės. Kuriant DB yra nurodomas jos vardas (pavadinimas), fizinė vieta, naudojama kalba ir kt. Kuriant lentelę - lentelės vardas, stulpelių vardai ir jų tipai.

SQL suprantami duomenų tipai: skaičiai (int, float, smallint...), tekstas (char, varchar, text, clob...), dvejetainiai duomenys (bit, bit varying, blob), datos ir laikai (date, time, timestamp).

DML sakiniais formuojamos užklausos (query) duomenims surasti, šalinti ir įterpti.

DCL sakiniais koordinuojamas vartotojų darbas su DB.

Papildoma informacija

www.sqlcourse.com

http://www.w3schools.com/sql/sql_intro.asp

https://academy.vertabelo.com

https://www.khanacademy.org/computing/computer-programming/sql

http://sqlzoo.net

https://www.tutorialspoint.com/sql/index.htm

https://www.sqlteaching.com/

Duomenų bazių naudojimas, SQL

domeną bazią nadaojimas, sąc

Duombazių naudojimas ir koregavimas

DML - duomenų valdymo sakiniai

- SELECT (išrinkti) duomenų išrinkimas iš vienos ar kelių lentelių;
- INSERT (jterpti) vieno ar kelių naujų įrašų įterpimas;
- UPDATE (atnaujinti) vieno ar kelių esamų įrašų atnaujinimas;
- DELETE (ištrinti) vieno ar kelių esamų įrašų ištrynimas;
- FROM (iš) iš kurių lentelių bus naudojami duomenys;
- WHERE (ten kur) sąlyga, kuri turią turi tenkinti naudojami duomenys;
- GROUP BY (sugrupuoti pagal) pagal kokį stulpelį(-ius) reikia grupuoti duomenis;
- ORDER BY (surikiuoti pagal) pagal kokį stulpelį(-ius) reikia rikiuoti duomenis;
- HAVING (turint) sąlyga, kurią turi tenkinti naudojami duomenys po grupavimo.

SQL SELECT sakinys

Naudojamas duomenų išrinkimui iš lentelės/lentelių.

Rezultatas gražinamas lentelės pavidalu, gauta lentelė vadinama "Result-Set".

Sintaksė:

 SELECT stulpelis1, stulpelis2,.. FROM lentelės1_pavadinimas, lentelės2_pavadinimas,..

arba

SELECT * FROM lentelės1_pavadinimas, lentelės2_pavadinimas,...

SQL SELECT sakinys

Pvz.:

SELECT * FROM DARBUOTOJAI;

olled Tricing by triboo 1 obtat,								
ASMENS_KODAS	VARDAS	PAVARDE	DIRBA_NUO	ATLYGINIMAS	PARDUOTUVES_ID			
38705203214	ALGIS	GRIGAS	2012-10-10	600.00	1			
39705203214	ANTANAS	PALIONIS	2013-12-10	500.00	2			
38905212217	KAROLIS	KARALIUS	2015-10-10	1100.00	3			
37805203245	GRIGAS	GRIGAITIS	2012-10-10	600.00	4			
47005203214	MARINA	GARINA	2013-10-10	300.00	2			
46605203214	VIKTORIJA	PETKEVIC	1999-10-10	800.00	1			

SELECT VARDAS, PAVARDE FROM DARBUOTOJAI - išrinktų tik stulpelių VARDAS ir PAVARDE duomenis iš lentelės DARBUOTOJAI.

SQL sąlygos sakinys WHERE

Naudojamas užduoti sąlygą išrenkamiems duomenims.

Sąlyga tikrinama kiekvienai lentelės(-ių) eilutei.

Pvz.: SELECT VARDAS, ATLYGINIMAS FROM DARBUOTOJAI WHERE VARDAS='Antanas' - išrinks vardą ir atlyginimą iš lentelės DARBUOTOJAI, kurių stulpelio VARDAS reikšmė lygi žodžiui Antanas.

SQL sintaksėje:

- 1. Žodžiai ir datos viengubose kabutėse ('Petras', '1991-12-11');
- 2. Stulpelių vardai neišskiriami arba dvigubose kabutėse;
- 3. Skaičiai neišskiriami.

SQL INSERT sakinys

Naudojamas naujo įrašo ar įrašų įterpimui į lentelę.

Sintaksė:

INSERT INTO lentelės_pavadinimas (stulpelis1,stulpelis2,...) VALUES (reikšmė1, reikšmė2,...), (reikšmė5, reikšmė6,...)

Stulpelių vardų galima ir nenurodyti, jei įterpiami duomenys užpildant visus lentelės stulpelius. Atitinkamai kai kuriuos laukus galima palikti neužpildytus.

Įvedamų reikšmių eiliškumas turi atitikti išvardytų stulpelių eiliškumą.

Pvz.: INSERT INTO DARBUOTOJAI (VARDAS) VALUES ('pabandukas')

SQL UPDATE sakinys

Naudojamas esamų įrašų duomenų atnaujinimui.

Sintaksė:

UPDATE lenteles_pavadinimas SET stulpelis1=reikšmė1, stulpelis2=reikšmė2 WHERE sąlyga

Nenurodžius sąlygos bus atnaujinti VISI įrašai.

PVZ.: UPDATE DARBUOTOJAI SET PAVARDE='pabandaitis' WHERE VARDAS='pabandukas'

Jei sąlyga atitinka kelias eilutes, jos visos bus atnaujintos.

SQL DELETE sakinys

Naudojamas esamų įrašų ištrynimui.

Sintaksė:

DELETE FROM lentelės_pavadinimas WHERE sąlyga

Ištrinamas visas įrašas. Jei nenurodoma sąlyga, ištrinti bus VISI lentelės įrašai.

Pvz.: DELETE FROM DARBUOTOJAI WHERE VARDAS='pabandukas'

Norint ištrinti visą lentelę kartu su jos struktūra, naudojamas DROP sakinys:

DROP TABLE lentelės_pavadinimas

SQL sąlygos sakinys WHERE (<>)

Sąlygą galima pateikti ir kaip aritmetinę išraišką arba simbolių eilutę, surištą per lyginimo operatorius =,>,<,>=,<=,<>(!=).

Pvz.: SELECT * FROM DARBUOTOJAI WHERE ATLYGINIMAS >= 600

SELECT VARDAS FROM DARBUOTOJAI WHERE PARDUOTUVES_ID <> 3

Sąlygoje galima lyginti ir dviejų stulpelių reikšmes.

SQL sąlygos sakinys WHERE (BETWEEN ... AND)

Reikšmių išrinkimas iš intervalo:

... WHERE stulpelis BETWEEN ... AND ... - jeigu reikalingos reikšmės iš intervalo, priimant ir jas pačias [imtinai].

Pvz.: SELECT VARDAS FROM DARBUOTOJAI WHERE ATLYGINIMAS BETWEEN 500 AND 700 - bus išrinkti vardai darbuotojų, kurių atlyginimai siekia nuo 500 iki 700.

Pirma nurodoma mažesnė reikšmė, po to didesnė.

SQL sąlygos sakinys WHERE (IN)

Reikalingų reikšmių sąrašo išvardinimas:

... WHERE stulpelis IN (...) - jei reikalingas ne intervalas, o tam tikros skirtingos reikšmės.

Pvz.: SELECT ASMENS_KODAS FROM DARBUOTOJAI WHERE VARDAS IN ('Algis', 'Grigas') - bus išrinkti visi Algių ir Grigų asmens kodai.

Reikšmių kiekis neribotas, galima įvesti ir vieną.

SQL sąlygos sakinys WHERE (LIKE)

Reikšmių išrinkimas pagal šabloną (panašumą):

... WHERE stulpelis LIKE ... - jei nėra žinomas tikslus duomuo arba reikalingos reikšmės, atitinkančios tik dalį duomens.

SQL pakaitos simboliai:

- % bet kokiam kiekiui bet kokių simbolių;
- vienam bet kokiam simboliui.

Pvz.: SELECT VARDAS FROM DARBUOTOJAI WHERE PAVARDE LIKE '%I_' - bus išvesti visi vardai darbuotojų, kurių priešpaskutinė pavardės raidė yra 'I'.

SQL sąlygos sakinys WHERE (NULL)

SQL egzistuoja reikšmė NULL - duomenys neįrašyti arba ištrinti.

Duomenų paieška pagal neįrašytas reikšmes:

... WHERE stulpelis IS NULL - jei reikia duomenų, kurie neturi reikšmės atitinkamame stulpelyje.

Pvz.: SELECT * FROM DARBUOTOJAI WHERE PARDUOTUVES_ID IS NULL - išrinks visus duomenis apie darbuotojus, kurie atleisti ar nepriskirti darbovietei. (PARDUOTUVES_ID neįrašytas.)

SQL sąlygos sakinys WHERE (NOT, AND, OR)

Duomenų paieška pagal kelias atitinkančias arba neatitinkančias sąlygas:

... WHERE sąlyga1 AND/OR sąlyga2 AND/OR NOT sąlyga3 - kai duomenys turi atitikti kelias sąlygas arba atitikti bent vieną arba neatitikti tam tikrų sąlygų.

Pvz.: SELECT * FROM DARBUOTOJAI WHERE ATLYGINIMAS > 300 AND PARDUOTUVES_ID = 2 - atitinkamai turi būti išpildytos abi sąlygos.

SELECT * FROM DARBUOTOJAI WHERE ATLYGINIMAS > 300 OR PARDUOTUVES_ID = 2 - jei išpildyta bent viena sąlyga, duomuo bus grąžintas.

SELECT * FROM DARBUOTOJAI WHERE ATLYGINIMAS > 300 AND NOT PARDUOTUVES_ID IN (2,3) - abi sąlygos turi būti išpildytos, antra paneigta.

SQL sąlygos sakinys WHERE (pirmenybės)

Lyginimo operacijos turi savo pirmenybes.

SQL atveju eiliškumo tvarka:

- 1. Visi lyginimo operatoriai (=,>,<,>=,<=,<>(!=));
- 2. NOT (paneigimas);
- 3. AND (ir);
- 4. OR (arba).

Pvz.: SELECT * FROM DARBUOTOJAI WHERE VARDAS = 'Algis' OR VARDAS = 'Karolis' AND ATLYGINIMAS = 1100

SELECT * FROM DARBUOTOJAI WHERE (VARDAS = 'Algis' OR VARDAS = 'Karolis') AND ATLYGINIMAS = 1100 - du skirtingi reikšmių rinkiniai.

SQL rikiavimo sakinys ORDER BY

Įprastai išvedamoje lentelėje duomenys yra rikiuojami pagal pirmą įvestą stulpelį (nuo A iki Z, nuo 1 iki 9).

Duomenų rikiavimas pagal pasirinktą(-us) stulpelį(-ius):

... ORDER BY stulpelis1, stulpelis2

Pvz.: SELECT * FROM DARBUOTOJAI ORDER BY VARDAS - surikiuota pagal vardus nuo A iki Z.

Norint rikiuoti atvirkščiai, pridedamas žodis DESC (descending):

SELECT * FROM DARBUOTOJAI ORDER BY VARDAS DESC

SQL kitos funkcijos

DISTINCT - išrenkamos skirtingos stulpelio reikšmės:

SELECT DISTINCT ATLYGINIMAS FROM DARBUOTOJAI - bus išrinkti skirtingi atlyginimai;

ILIKE - LIKE atmaina šablonams, kurie nejautrūs didžiosioms/mažosioms raidėms:

SELECT * FROM DARBUOTOJAI WHERE VARDAS ILIKE 'algis' - išrinks ir ALGIUS, ir Algius, ir algius.

SQL kitos funkcijos

LOWER(stulpelis/reikšmė) - bet kokius formatu parašytus duomenis pateiks mažosiomis raidėmis:

SELECT LOWER (VARDAS) FROM DARBUOTOJAI - darbuotojų vardadi bus pateikti mažosiomis raidėmis

UPPER(stulpelis/reikšmė) - LOWER atitikmuo didžiosioms raidėms.

CONCAT - dviejų ar daugiau stulpelių reikšmių formatuotas išspausdinimas viename stulpelyje:

SELECT CONCAT (VARDAS, ' ir ', PAVARDE) FROM DARBUOTOJAI - bus išspausdinti vardai ir pavardės darbuotojų su žodžiu 'ir' tarp jų.

SQL kitos funkcijos

Skaičiuotuvas:

SELECT ATLYGINIMAS*10 FROM DARBUOTOJAI

SELECT VARDAS, 8*13 FROM DARBUOTOJAI;

SELECT 5+3;

SELECT PARDUOTUVES_ID/3 FROM DARBUOTOJAI;

SQL grupavimo sakinys GROUP BY

Pagal nutylėjimą visos lentelės eilutės skaitomos kaip viena grupė.

Norint sudalinti visas eilutes arba jų dalį, SELECT sakinyje naudojama GROUP BY komanda (jei norime suskirstyti duomenis pagal tam tikrą aspektą).

Grupėms gali būti naudojamos grupinės funkcijos:

- 1. AVG() nustato vidurkį grupei;
- 2. COUNT() nustato eilučių, kurių išraiška yra apibrėžta, kiekį lentelėje;
- 3. MAX() nustato didžiausią reikšmę;
- 4. MIN() nustato mažiausią reikšmę;
- 5. SUM() nustato bendrą sumą, ignoruojant neapibrėžtas reikšmes.

SQL grupavimo sakinys GROUP BY

Įprastu atveju grupavimas ignoruoja NULL reikšmes, nebent naudojamas visų eilučių skaičiavimas COUNT(*).

Grupavimo pavyzdys:

SELECT AVG(ATLYGINIMAS) FROM DARBUOTOJAI - išves bendrą atlyginimo vidurkį visiems.

SELECT AVG(ATLYGINIMAS),PARDUOTUVES_ID FROM DARBUOTOJAI GROUP BY PARDUOTUVES_ID - išves atlyginimų vidurkius kiekvienoje parduotuvėje.

SQL grupavimo sakinys GROUP BY

SELECT COUNT(*) FROM DARBUOTOJAI WHERE ATLYGINIMAS = 600 - išves įrašų kiekį lentelėje, atitinkantį sąlygą

SELECT COUNT(*),PARDUOTUVES_ID FROM DARBUOTOJAI WHERE ATLYGINIMAS = 600 GROUP BY PARDUOTUVES_ID - išves darbuotojų kiekį, atitinkantį sąlygą, kiekvienoje parduotuvėje.

Galima grupuoti ir pagal kelis stulpelius:

SELECT SUM(ATLYGINIMAS) FROM DARBUOTOJAI GROUP BY PARDUOTUVES_ID, ATLYGINIMAS - toje pačioje parduotuvėje tokį patį atlyginimą gaunančių darbuotojų išmokėtos atlyginimų sumos.

SQL sąlygos sakinys HAVING grupėms

Sąlygos sakinio WHERE negalima naudoti sugrupuotiems duomenims, jis naudojamas duomenų atrinkimui prieš grupavimą. Norint užduoti sąlygą grupuotiems duomenims, naudojamas HAVING sakinys.

Pvz.: SELECT COUNT(*),PARDUOTUVES_ID FROM DARBUOTOJAI WHERE ATLYGINIMAS = 600 GROUP BY PARDUOTUVES_ID HAVING COUNT(*) >= 2 - išves parduotuvės ID ir skaičių tų darbuotojų parduotuvėse, kurių atlyginimas yra 600 ir tas skaičius yra didesnis arba lygus 2.

Bendra SQL sintaksė

SELECT stulpelis1, stulpelis2, grupine_funkcija

FROM lentelė

WHERE sąlyga

GROUP BY sąrašas grupavimui

HAVING sąlygai grupei

ORDER BY rikiavimo stulpelis

Duomenų bazių naudojimas, SQL

Įvadas į projektavimą

Lentelių kūrimas

Lentelės kuriamos CREATE TABLE sakiniu:

CREATE TABLE lentelės_pavadinimas (stulpelis1 tipas1, stulpelis2 tipas2,..)

Duombazių priimamos duomenų rūšys:

- Simboliniai duomenys;
- Dvejetainiai duomenys;
- Skaičiai;
- Datos;
- Laikas.

Simboliniai duomenys (character datatypes)

Simbolių eilutė gali priimti tam tikrą kiekį simbolių.

Fiksuoto ilgio simbolių eilutės:

CHAR(n) - n iki 254;

Kintamo ilgio:

- VARCHAR(n) n iki 2 147 483 647 ilgio;
- CLOB(n[K|M|G]) iki 2 GB (4 GB).

Kai n > 254, užklausose negalima naudoti: DISTINCT, GROUP BY, ORDER BY

Skaitiniai duomenys (number datatypes)

- SMALLINT sveikieji skaičiai [-32 768; 32 767];
- INTEGER "dideli" sveikieji skaičiai (4 baitai) [-2 147 483 648; 2 147 483 647];
- BIGINT "ypač dideli" sveikieji skaičiai (8 b);
- REAL slankiojo kablelio skaičiai, 32 baitai, pvz., -2E5, 5.555E-18,
 -.655645e8;
- FLOAT (DOUBLE) dvigubo tikslumo skaičiai (64 b);
- DECIMAL (n, m) (NUMERIC) dešimtainiai supakuoti skaičiai iki 1000 (31) dešimtainių skaitmenų.

Dvejetainiai duomenų tipai (binary datatypes).

Fiksuoto ilgio:

BIT(n).

Kintamojo ilgio:

- BIT VARYING(n);
- BLOB(n[K|M|G]).

Datos ir laiko duomenys

- DATE (atmintyje 4 baitai);
- TIME (3 baitai);
- TIMESTAMP (10 baitų).

DATE ir TIME vaizdavimas priklauso nuo terpės (locale):

2005.01.01, 12:00:00,

2005-01-01-12.15.55.330000.

DBVS visada "supranta" ISO datą ir laiką:

2005-01-01, 12:00:00.

Lentelių kūrimas

Pvz. darbuotojų lentelei:

CREATE TABLE DARBUOTOJAI (ASMENS_KODAS BIGINT NOT NULL, VARDAS VARCHAR(20), PAVARDE VARCHAR(30), DIRBA_NUO DATE, ATLYGINIMAS DECIMAL (7,2) DEFAULT 500, PARDUOTUVES_ID SMALLINT)

NOT NULL - stulpelis negali įgyti NULL reikšmių.

DEFAULT - reikšmė pagal nutylėjimą (kokią įgaus, jei duomenys nebus įterpiami į stulpelį)

Lentelių kūrimas

Susikurkime lentelę parduotuvėms:

CREATE TABLE PARDUOTUVES (ID INT, MIESTAS VARCHAR(20), ADRESAS VARCHAR (40))

Galimos parinktys stulpeliams:

AUTO_INCREMENT - automatinis didėjimas (nuo 1);

UNIQUE - stulpelio reikšmės būtinai skirsis viena nuo kitos, nebus pasikartojimų.

Lentelių kūrimas

Daugumą lentelės savybių galima nurodyti vėliau naudojant ALTER sakinį (lentelės struktūros keitimas):

ALTER TABLE DARBUOTOJAI ADD Gimtadienis DATE

ALTER TABLE DARBUOTOJAI DROP Gimtadienis

ALTER TABLE PARDUOTUVES ALTER MIESTAS SET DEFAULT 'Vilnius'

ALTER TABLE PARDUOTUVES ALTER Svarba DROP DEFAULT

Vykdytojai

Nr	Pavardė	Kvalifikacija	Kategorija	Išsilavinimas
1	Jonaitis	Informatikas	2	VU
2	Petraitis	Statistikas	3	VU
3	Gražulytė	Inžinierius	1	NULL
4	Onaitytė	Vadybininkas	5	VDU
5	Antanaitis	Informatikas	3	VU

Projektai

Nr	Pavadinimas	Svarba	Pradžia	Trukmė
1	Studentų apskaita	Aukšta	2005.01.01	12
2	Buhalterinė apskaita	Vidutinė	2005.03.01	10
3	WWW svetainė	Ypatinga	2005.06.01	2

Vykdymas

Projektas	Vykdytojas	Statusas	Valandos
1	1	Programuotojas	30
1	2	Dokumentuotojas	100
1	3	Testuotojas	100
1	4	Vadovas	100
2	1	Programuotojas	300
2	2	Analitikas	250
2	4	Vadovas	100
3	1	Programuotojas	250
3	2	Vadovas	400
3	3	Dizaineris	150

Lentelių jungimas (WHERE)

Reliacinis modelis pagrįstas duomenų skaidymu į lenteles, jas susiejant per bendrą lauką.

Norint rasti duomenis apie objektą, kai tie duomenys yra išskaidyti per kelias lenteles, naudojamas lentelių jungimas.

Sintaksė:

SELECT lentelė1.stulpelis1, lentelė2.stulpelis2 FROM lentelė1, lentelė2 WHERE jungimo_sąlyga

Jungimo sąlyga - tai stulpelis, laikantis bendrus duomenis ir pirmoje, ir antroje lentelėse.

Kitas būdas jungti lenteles:

SELECT stulpelis1, stulpelis2 FROM lentelė1 JOIN lentelė2 ON jungimo_sąlyga

Naudojant JOIN lengviau atskirti jungimą nuo sąlygų. JOIN turi ir kitų galimybių.

Galima jungti ir daugiau lentelių:

SELECT stulpelis1, stulpelis2, stulpelis3 FROM lentelė1, lentelė2, lentelė3 WHERE jungimo_1_ir_2_sąlyga AND jungimo_2_ir_3_sąlyga

SELECT stulpelis1, stulpelis2, stulpelis3 FROM lentelė1 JOIN lentelė2 ON jungimo_1_ir_2_sąlyga JOIN lentelė3 ON jungimo_2_ir_3_sąlyga

Lenteles galima jungti su jomis pačiomis. Pavardės vykdytojų, turinčių vienodą kvalifikaciją:

SELECT A.Pavarde, B.Pavarde FROM Vykdytojai A JOIN Vykdytojai B ON A.Kvalifikacija=B.Kvalifikacija WHERE A.Nr <> B.Nr

A.Nr <> B.Nr reikalinga išmesti visus duplikatus.

Visų projektų vykdytojų pavardės, statusai ir valandos:

SELECT Pavarde, Statusas, Valandos FROM Vykdytojai JOIN Vykdymas ON Vykdytojai.Nr=Vykdymas.Vykdytojas

Užklausos rezultate 5-ojo vykdytojo nebus, jis nedalyvauja jokiame projekte, jo nėra lentelėje vykdymas.

Visi vykdytojai gali būti pateikti taikant išorinį jungimą.

LEFT OUTER JOIN - rezultatas papildomas kairiosios lentelės nesujungiamomis eilutėmis.

RIGHT OUTER JOIN - rezultatas papildomas dešiniosio lentelės nesujungiamomis eilutėmis.

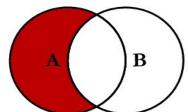
FULL OUTER JOIN - kai kuriose DBVS veikiantis jungimas, rezultatas papildomas iš abiejų pusių lentelių nesujungiamomis eilutėmis.

JOIN atitinka INNER JOIN jungimą.

A B

SQL JOINS

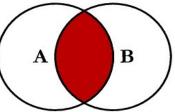
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key

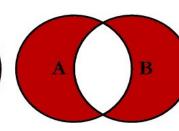


SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key

SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL

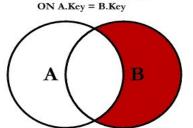
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key





A B

SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B



SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL

SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL

Darbuotojai ir jų dalyvavimas projekte Nr 1.:

SELECT Pavarde, Statusas, Valandos FROM Vykdytojai LEFT OUTER JOIN Vykdymas ON Nr=Vykdytojas WHERE Projektas = 1

Arba:

SELECT Pavarde, Statusas, Valandos FROM Vykdymas RIGHT OUTER JOIN Vykdytojai ON Nr=Vykdytojas WHERE Projektas = 1

Rezultatas tas pats.

Struktūrinės užklausos

Vienoje užklausoje gali būti ir kita užklausa.

Pavardės vykdytojų, dalyvaujančių projekte Nr. 1:

SELECT Pavarde FROM Vykdytojai WHERE Nr IN (SELECT Vykdytojas FROM Vykdymas WHERE Projektas = 1.)

Vykdytojai, dalyvaujantys bent viename ypatingos svarbos projekte:

SELECT Pavarde FROM Vykdytojai WHERE Nr IN (SELECT Vykdytojas FROM Vykdymas WHERE Projektas IN (SELECT Nr FROM Projektai WHERE Svarba = 'Ypatinga'))

Pirminis raktas (Primary Key)

Pirminis raktas - tai stulpelio ar stulpelių apribojimas, neleidžiantis jam ar jiems įgyti pasikartojančių reikšmių ir reikšmės NULL. Pvz.:

CREATE TABLE TEST(ID INT PRIMARY KEY) - ID neįgis NULL, nebus pasikartojantis.

ALTER TABLE PARDUOTUVES ADD PRIMARY KEY (P_Id)
ALTER TABLE PARDUOTUVES ADD CONSTRAINT pk_Miest_Adres PRIMARY
KEY (MIESTAS, ADRESAS)

ALTER TABLE PARDUOTUVES DROP CONSTRAINT pk_Miest_Adres

Negalima apriboti stulpelio pirminiu raktu, jei jis priima NULL reikšmes.

Antrinis raktas (Foreign Key)

Antrinis raktas - tai stulpelio apribojimas, neleidžiantis stulpelio reikšmėms išeiti iš reikšmių rėžių, kurios yra kitoje lentelėje pirminio rakto stulpelyje.

ALTER TABLE DARBUOTOJAI ADD FOREIGN KEY (PARDUOTUVES_ID) REFERENCES PARDUOTUVES(ID)

ALTER TABLE DARBUOTOJAI ADD CONSTRAINT fk_DarPard FOREIGN KEY (PARDUOTUVES_ID) REFERENCES PARDUOTUVES(ID)

ALTER TABLE DARBUOTOJAI DROP CONSTRAINT fk_DarPard

DB projektavimas

Atliekamas prieš pradedant DB formavimo procesą.

Svarbus norint sukurti efektyvią, tikslią, poreikius atitinkančią sistemą.

Patartina planavimą pradėti raštu ant balto popieriaus lapo (MS Access).

Nors atrodo, kad toks projektavimas yra laiko gaišimas pradžioje, tačiau iš tiesų daugiau laiko sugaištama vėliau, jei prieš tai neplanuojama.

DB projektavimo procesas

- 1. Numatyti duomenų bazės paskirtį;
- 2. Apibrėžti, kokios lentelės bus reikalingos;
- 3. Apibrėžti, kokie laukai turės būti DB lentelėse;
- 4. Nurodyti laukus su unikaliomis reikšmėmis;
- 5. Nustatyti, kaip lentelės siesis tarpusavyje;
- 6. Tobulinti pradinį projektą.

DB paskirtis

Pradedant kurti duomenų bazę reikia atsakyti į tris esminius klausimus:

- Kokius duomenis turėtų kaupti duomenų bazė?
- 2. Kokią informaciją apie šiuos duomenis norės žinoti DB vartotojas?
- 3. Kokius veiksmus vartotojas norės atlikti su duomenimis?

DB paskirtis

Pavyzdys: duomenų bazė bibliotekai.

- 1. Duomenų bazėje turėtų būti duomenys apie skaitytojus ir knygas.
- 2. DB vartotojas norės gauti informaciją apie tai, kokias knygas kas skaito, kada turi grąžinti, kiek kokio autoriaus knygų yra bibliotekos fonde ir pan.
- 3. Vartotojas norės iš DB gauti informaciją, taip pat ją pakeisti, pasikeitus situacijai grąžinamos ar paskolinamos knygos, nauji skaitytojai, nauji leidiniai, ištrinti įrašus apie nurašytas knygas ir pan.

Lentelių apibrėžimas

Apibrėžus paskirtį, apibrėžiamos lentelės:

- Apie kokius objektus bus kaupiami duomenys?
- Siektina, kad visa informacija apie objektą būtų toje pačioje lentelėje, o kiekviena lentelė būtų skirta tik vienam objektui - ne visada tai pavyksta.

Lentelių apibrėžimas

Bibliotekos pavyzdys:

- Informacija bus kaupiama apie skaitytojus, autorius, knygas bei apie tai, kas kokią knygą yra pasiskolinęs;
- 4 lentelės:

Skaitytojai

Autoriai

Knygos

"Nuomos sutartys"

Nusprendus, kokiose lentelėse bus kaupiami duomenys, reikia nuspręsti, kokia informacija bus kaupiama apie kiekvieną objektą.

Jos turi būti būtent tiek, kiek reikalinga, nei per daug, nei per mažai.

Skaitytojai:



Autoriai:

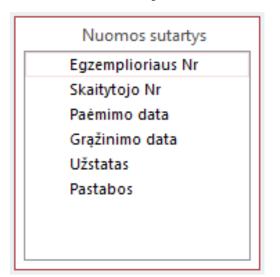


Knygos:

Knygos

Egzemplioriaus nr
Pavadinimas
Žanras
Autoriaus Nr
Leidimo metai
ISBN
Fizinė būklė

Nuomos sutartys:



Formuojant DB lenteles ir laukus reikėtų prisilaikyti šių principų:

- Kiekviena informacijos detalė saugoma tik vienoje lentelėje;
- Lentelėje neturėtų būti pasikartojančios informacijos;
- Kiekvienoje lentelėje informacija turėtų būti tik apie vieną objektą.

Pirminiai raktai

Kiekviena lentelė turėtų turėti savo pirminį raktą – lauką, kurio reikšmė kiekvienam įrašui yra unikali.

Pirminis raktas gali būti:

Vieno lauko - automatiškai didėjantis (1,2,3... a,b,c) arba vartotojo nustatomas (asmens kodas, ISBN, studento pažymėjimo nr ir pan)

Daugelio laukų (multi-field) - unikali ne konkretaus lauko reikšmė, bet keleto laukų reikšmių rinkinys (vardas IR pavardė, skaitytojo nr. IR egzemplioriaus nr.)

Išoriniai raktai

Pirminiai raktai taip pat atlieka vaidmenį kuriant ryšius tarp lentelių. Vienos lentelės pirminis laukas įterpiamas į kitą lentelę ir tampa ten išoriniu raktu.



DB normalizavimas

DB normalizavimas - tai patikrinimas, kad:

- 1. Lentelėje yra tik su pirminiu raktu susiję duomenys;
- 2. Kiekviename lauke yra tik vienas duomenų blokas;
- 3. Pasikartojantys duomenys yra eliminuoti.

Normalizuojant duomenys yra priskiriami ten, kur yra jų logiška ir unikali vieta.

DB normalizavimas

Normalizavimo tikslai:

- Duomenų surikiavimas į tokias logines grupes, kad kiekviena grupė apibūdintų dalį visumos;
- 2. Pasikartojančių duomenų kiekio duomenų bazėje minimizavimas;
- Efektyviai ir greitai prieinamos ir valdomos duomenų bazės sukūrimas nepakenkiant duomenų vientisumui;
- 4. Sukūrimas tokios struktūros, kad keičiant duomenis, juos reikėtų taisyti tik vienoje vietoje.

DB normalizavimas

Normalinės formos:

- Nenormalizuoti duomenys pasikartojančios grupės, netolygūs duomenys, įterpimo ir trynimo anomalijos.
- 1 normalinė forma kiekvienas DB laukas saugo tik vieną reikšmę ir lentelėje nėra pasikartojančių grupių.
- 2 normalinė forma atitinka 1 formos reikalavimus ir tiesiogiai nuo pirminio rakto nepriklausoma informacija turi būti perkelta į kitą lentelę.
- 3 normalinė forma atitinka 2 formos reikalavimus ir visi laukai, kurių reikšmės gali būti išgautos iš kitų lentelės laukų duomenų, turi būti pašalinti.

1 normalinė forma (blogas variantas)

EmployeeID	Name	Project	Time
EN1-26	Sean O'Brien	30-452-T3, 30-457-T3, 32-244-T3	0.25, 0.40, 0.30
EN1-33	Amy Guya	30-452-T3, 30-382-TC, 32-244-T3	0.05, 0.35, 0.60
EN1-35	Steven Baranco	30-452-T3, 31-238-TC	0.15, 0.80
EN1-36	Elizabeth Roslyn	35-152-TC	0.90
EN1-38	Carol Schaaf	36-272-TC	0.75
EN1-40	Alexandra Wing	31-238-TC, 31-241-TC	0.20, 0.70

EmpID	Last Name	First Name	Project1	Time1	Project2	Time2	Project3	Time3
EN1-26	O'Brien	Sean	30-452-T3	0.25	30-457-T3	0.40	32-244-T3	0.30
EN1-33	Guya	Amy	30-452-T3	0.05	30-382-TC	0.35	32-244-T3	0.60
EN1-35	Baranco	Steven	30-452-T3	0.15	31-238-TC	0.80		
EN1-36	Roslyn	Elizabeth	35-152-TC	0.90				
EN1-38	Schaaf	Carol	36-272-TC	0.75				
EN1-40	Wing	Alexandra	31-238-TC	0.20	31-241-TC	0.70		

1 normalinė forma (tenkinantis variantas)

EmployeeID	Last Name	First Name	
EN1-26	O'Brien	Sean	
EN1-33	Guya	Amy	
EN1-35	Baranco	Steven	
EN1-36	Roslyn	Elizabeth	
EN1-38	Schaaf	Carol	
EN1-40	Wing	Alexandra	

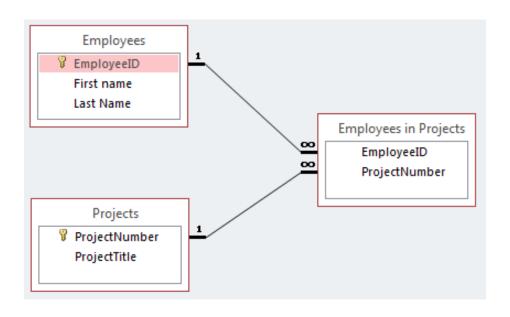
PROJECTS_EMPLOYEES_TIME

ProjectNum	EmployeeID	Time
30-328-TC	EN1-33	0.35
30-452-T3	EN1-26	0.25
30-452-T3	EN1-33	0.05
30-452-T3	EN1-35	0.15
31-238-TC	EN1-35	0.80
30-457-T3	EN1-26	0.40
31-238-TC	EN1-40	0.20
31-241-TC	EN1-40	0.70
32-244-T3	EN1-33	0.60
35-152-TC	EN1-36	0.90
36-272-TC	EN1-38	0.75

2 normalinė forma (blogas variantas)

*EmployeeID	LastName	FirstName	*ProjectNumber	ProjectTitle
EN1-26	O'Brien	Sean	30-452-T3	STAR manual
EN1-26	O'Brien	Sean	30-457-T3	ISO procedures
EN1-26	O'Brien	Sean	31-124-T3	Employee handbook
EN1-33	Guya	Amy	30-452-T3	STAR manual
EN1-33	Guya	Amy	30-482-TC	Web Site
EN1-33	Guya	Amy	31-241-TC	New catalog
EN1-35	Baranco	Steven	30-452-T3	STAR manual
EN1-35	Baranco	Steven	31-238-TC	STAR prototype
EN1-36	Roslyn	Elizabeth	35-152-TC	STAR pricing
EN1-38	Schaaf	Carol	36-272-TC	Order system
EN1-40	Wing	Alexandra	31-238-TC	STAR prototype
EN1-40	Wing	Alexandra	31-241-TC	New catalog

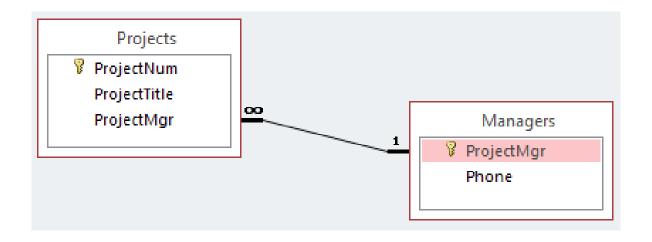
2 normalinė forma (tenkinantis variantas)



3 normalinė forma (blogas variantas)

*ProjectNum	ProjectTitle	ProjectMgr	Phone
30-452-T3	STAR manual	Garrison	2756
30-457-T3	ISO procedures	Jacanda	2954
30-482-TC	Web site	Friedman	2846
31-124-T3	Employee handbook	Jones	3102
31-238-TC	STAR prototype	Garrison	2756
31-241-TC	New catalog	Jones	3102
35-152-TC	STAR pricing	Vance	3022
36-272-TC	Order system	Jacanda	2954

3 normalinė forma (tenkinantis variantas)



Ryšiai tarp lentelių

Ryšys – sąsaja tarp bendrų laukų dvejose duomenų bazės lentelėse.

Trys tipai:

- 1:1 naudojamas retai, paprastai dėl saugumo, arba kai dalis lentelės laukų dažniausiai būna tušti (pvz. Mokinių lankomumas ir anketų pildymas);
- 1:∞ dažniausiai naudojamas ryšys. Pirminis raktas su išorinio rakto lauku kitoje lentelėje;
- ∞:∞ reikia skaidyti į du 1:∞ ryšius per tarpinę lentelę (netraktuojamas kaip tikras ryšys).

Tobulinimas

Sudėliojus pirminį DB projektą (lentelės, laukai ir ryšiai) reikia atidžiai jį išnagrinėti, ieškant galimų trūkumų.

Patikrinkite, ar tokia duomenų bazės struktūra leis jums gauti tuos atsakymus, kurių norite.

Paskutinė galimybė lengvai ištaisyti DB struktūros klaidas:

- Per mažai dėmesio planavimui;
- Normalizavimo ignoravimas;
- Prastai parinkti laukų/lentelių pavadinimai;
- Dokumentacijos trūkumas.