SQL

SELECT
[DISTINCT | ALL] < column-list>
FROM < table-names>
[WHERE < condition>]
[ORDER BY < column-list>]
[GROUP BY < column-list>]
[HAVING < condition>]

[]- не е задължително /- или

SQL

ORDER BY

Клаузата ORDER BY сортира, подрежда резултатите от заявката

Можете да подреждате във възходящ и низходящ ред Може да се изпълни за няколко колони

Не може да се изпълни за колона, която не е в резултатната таблица

SELECT <columns> FROM <tables>WHERE <condition>
ORDER BY <cols>[ASCENDING |DESCENDING|ASC | DESC]

SQL Пример: SELECT * FROM Grades ORDER BY Mark Grades ORDER BY Mark 112 Иван Стоян 125 Иван 125 137 112 Милена Иван 112 Мария 136 125 Мария 137 Милена 112 5 Стоян 125 Мария 136

SQL Пример: SELECT * FROM Grades ORDER BY Code ASC, Mark DESC Grades ORDER BY Mark Code ASC, Mark DESC Иван 112 Милена 112 Иван 125 Иван 112 Милена 112 Иван 125 Мария 136 Стоян 125 Мария 137 Мария 136 Стоян 125 Мария 137

SQL

Kонстанти и аритметика

Освен с имена на колони, чрез SELECT може да се оперира с константи, да се извършват аритметични действия и изчисляват функции.

SELECT 1.5*Mark FROM Grades
SELECT Salary + Bonus FROM Employee
SELECT Price /10 FROM Products

яде

Агрегатни функции

Агрегатните функции изчисляват обобщения върху данните от таблицата

Повечето агрегатни функции (всички освен COUNT) изчисляват върху една колона с числови данни Използва се псевдоним за резултата

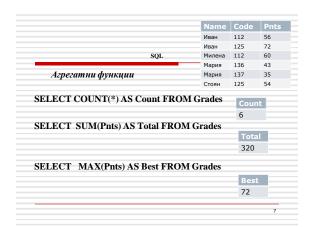
Агрегатни функции:

СОИNТ: Брой на редовете

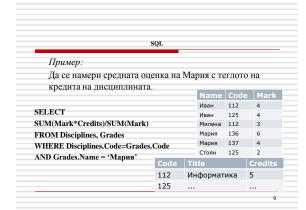
SUM: Сумата от една колона

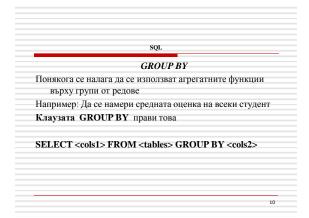
AVG: Средноаритметичната стойност в една колона

МIN, МАХ: Минималната и максималната стойност в една колона



	Иван Иван	112 125	56 72
SQL	Милена	112	60
	Мария	136	43
Агрегатни функции	Мария	137	35
	Стоян	125	54
Чрез аритметични действия м агрегатните функции SELECT MAX(Pnts)-MIN(I		•	
агрегатните функции		•	
агрегатните функции SELECT MAX(Pnts)-MIN(I	Pnts) AS Range	•	
агрегатните функции SELECT MAX(Pnts)-MIN(I Grades	Pnts) AS Range	FROM	
arperarните функции SELECT MAX(Pnts)-MIN(I Grades MAX(Pnts) = 72	Pnts) AS Range	FROM	





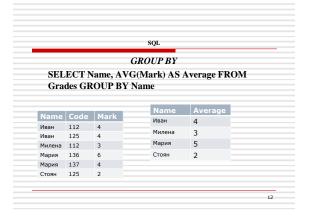
sql

GROUP BY

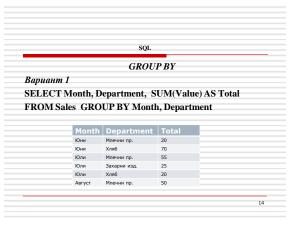
SELECT <cols1> FROM <tables> GROUP BY <cols2>

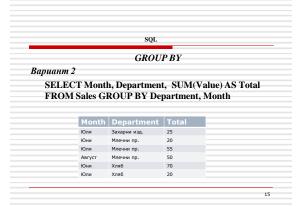
Всеки елемент от <cols2> трябва да присъства в <cols1>

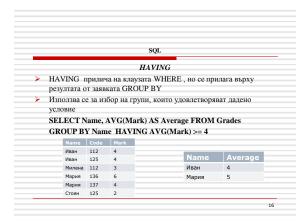
Може да има WHERE или ORDER BY клаузи











у WHERE и HAVING

✓ WHERE се съотнася към редове от таблици и поради това не използва агрегатни функции

✓ HAVING се съотнася към групи от редове от таблици и поради това не може да използва колони, които не са включени в GROUP BY

 SQL

 WHERE и HAVING

 Обработка на заявката:
 • Таблиците се комбинират (...)

 • WHERE клаузи
 • GROUP BY и агрегатни функции

 • Избор на колони
 • HAVING клаузи

 • ORDER BY
 • ORDER BY

устануварания в предменять и предменять пре



SQL

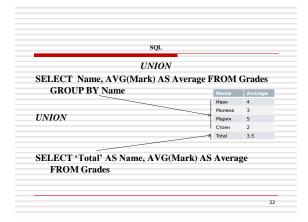
UNION

Заявката за средноаритметичната оценка на всеки студент:

SELECT Name, AVG(Mark) AS Average FROM Grades GROUP BY Name

Заявката за средноаритметичната оценка на всички студенти:

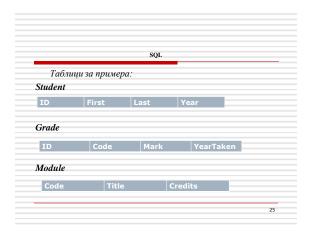
SELECT 'Total' AS Name, AVG(Mark) AS Average FROM Grades

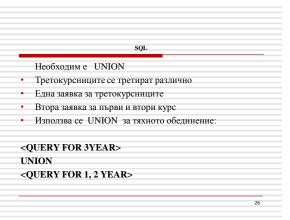


уст.

Пример:
Заявки за:
Списък на студентите и техните средни оценки
За студентите от първи и втори курс – техните средни оценки за годината
За студентите от трети курс да се пресметне 40% от средния успех на втората година изчислено с 60% на средния успех от третата година.

Резултатите да са:
Сортирани по година, след това средната оценка (от по-висока към по-ниска), после първото име и накрая ID
Да се вземе предвид броя на кредитите на всяка дисциплина
Да се реализира с една заявка





устверення в развительный в развите

SQL

3arbrata до този момент

SELECT <some information>
FROM Student, Module, Grade
WHERE Student.ID = Grade.ID
AND Module.Code = Grade.Code
AND <student is in 3 year>
UNION
SELECT <some information>
FROM Student, Module, Grade
WHERE Student.ID = Grade.ID
AND Module.Code = Grade.Code
AND <student is in 1 or 2 year>

устверения в короматия в курс

Упрябва да се извлекат:

Изчисляване на средноаритметична оценка, с тегло 40-60 съответно за 2-ри и 3-ти курс

Оценките от първата година се игнорират

Нужни са ID, Name и Year за подредбата

Средноаритметична оценка е проблемна

Няма как директно и лесно да се разделят оценките от 2 и 3 курс

Би могло да се използва, че 40 = 20*2 и 60 = 20*3, така че между YearTaken (завършения курс) и теглото на оценките да има някаква зависимост

SQL

3asekama 3a 3-mu kypc

SELECT Year, Student.ID, Last, First,

SUM((20*YearTaken/100)*Mark*Credits)/120

AS AverageMark

FROM Student, Module, Grade

WHERE Student.ID = Grade.ID

AND Module.Code = Grade.Code

AND YearTaken IN (2,3)

AND Year = 3

GROUP BY Year, Student.ID, First, Last

ус.

Информация за 1-ви и 2-ри курс

Трябва само да се вземат средноаритметичните оценки, където YearTaken и Year са с една и съща стойност

Нужни са ID, Name и Year за подредбата, както в другата заявка

SQL

3asekama 3a 1-eu u 2-pu kypc

SELECT Year, Student.ID, Last, First,
SUM(Mark*Credits)/120 AS AverageMark
FROM Student, Module, Grade
WHERE Student.ID = Grade.ID
AND Module.Code = Grade.Code
AND YearTaken = Year
AND Year IN (1,2)
GROUP BY Year, Student.ID, First, Last

Финалната заявка

SELECT Year, Student.ID, Last, First,
SUM((20*YearTaken/100)*Mark*Credits)/120 AS AverageMark
FROM Student, Module, Grade
WHERE Student.ID = Grade.ID AND Module.Code = Grade.Code
AND YearTaken IN (2,3) AND Year = 3
GROUP BY Year, Student.ID, First, Last
UNION
SELECT Year, Student.ID, Last, First,
SUM(Mark*Credits)/120 AS AverageMark
FROM Student, Module, Grade
WHERE Student.ID = Grade.ID AND Module.Code = Grade.Code
AND YearTaken = Year AND Year IN (1,2)
GROUP BY Year, Student.ID, First, Last
ORDER BY Year desc, AverageMark desc, First, Last, ID

