

## SQL

## SQL SELECT

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

[]- не е задължително /- или

1

## SQL

## Пример: Student

ID	First	Last
12101114	Ангелина	Пеева
12101115	Боян	Кръстев
12101116	Венета	Стоилова
12101117	Георги	Пешев
12101118	Ангел	Кръстев

## Grade

ID	Code	Mark
12101114	112	4
12101114	125	3
12101115	112	5
12101115	136	5
12101117	172	3
12101118	112	6
12101118	125	5
12101118	136	4

## Course

Code	Title
112	Информатика
125	Математика
136	Физика
172	ПИК 1

2

## SQL

## SQL SELECT - DISTINCT и ALL

- В една колона може да има повтарящи се стойности
- Използването на **DISTINCT** премахва дублираните стойности
- Използването на **ALL** ги възстановява и е по подразбиране

Last	← SELECT ALL Last FROM Student	Last
Пеева		Пеева
Кръстев		Кръстев
Стоилова		Стоилова
Пешев		Пешев
Кръстев		Кръстев

3

## SQL

## SQL SELECT с клаузата WHERE

- Клаузата **WHERE** ограничава редовете, които ще се върнат
- Тази клауза задава условие – само тези редове, които отговарят на условието ще се върнат
- Примерни условия:
  - Mark < 4
  - First = 'Боян'
  - First <> 'Боян'
  - (First = 'Боян') AND (Last = 'Кръстев')
  - (Mark < 3) OR (Mark > 5)

4

## SQL

## Примери:

SELECT \* FROM Grade WHERE Mark >= 5

ID	Code	Mark
12101115	112	5
12101115	136	5
12101118	112	6
12101118	125	5

SELECT DISTINCT ID FROM Grade WHERE Mark >= 5

ID
12101115
12101118

5

## SQL

## Пример:

## Grade

Напишете SQL заявка за намиране на списък на ID номерата на студентите и оценките им по Информатика, ако тези оценки са по-високи от 4.

ID	Code	Mark
12101115	112	5
12101118	112	6

6

## SQL

Решение 1 :

**SELECT ID, Mark FROM Grade WHERE (Code = 'Информатика') AND (Mark > 4)**

7

## SQL

**SELECT** от няколко таблици

- Често е необходимо
- **SELECT \* FROM Table1, Table2...**
- Ако таблиците имат колони с еднакви имена ...
- Това се решава чрез рефериране на колоните с името на таблицата:

**TableName.Column**

8

## SQL

**SELECT \* FROM Table1, Table2...**

(без WHERE) При Table1 - 20 реда и Table2 - 20 реда – резултат?

Резултатът е:

1ред Table1      1ред Table2  
1ред Table1      2ред Table2  
1ред Table1      3ред Table2

.....  
2ред Table1      1ред Table2  
2ред Table1      2ред Table2  
2ред Table1      3ред Table2  
.....

9

## SQL

Пример: Student

Grade

ID	First	Last	ID	Code	Mark
12101114	Ангелина	Пеева	12101114	112	4
12101114	Ангелина	Пеева	12101114	125	3
12101115	Боян	Кръстев	12101115	112	5
12101116	Венета	Стоилова	12101115	136	5
12101117	Георги	Пешев	12101117	172	3
12101118	Ангел	Кръстев	12101118	112	6
			12101118	125	5
			12101118	136	4

**SELECT First, Last, Mark FROM Student, Grade  
WHERE (Student.ID =Grade.ID) AND (Mark > 4)**

Еквивалентно и нееквивалентно свързване.

10

## SQL

**SELECT ... FROM Student, Grade WHERE...**

ID	First	Last	ID	Code	Mark
12101114	Ангелина	Пеева	12101114	112	4
12101114	Ангелина	Пеева	12101114	125	3
12101114	Ангелина	Пеева	12101115	112	5
12101114	Ангелина	Пеева	12101115	136	5
12101114	Ангелина	Пеева	12101117	172	3
12101114	Ангелина	Пеева	12101118	112	6
12101114	Ангелина	Пеева	12101118	125	5
12101114	Ангелина	Пеева	12101118	136	4
12101115	Боян	Кръстев	12101114	112	4
12101115	Боян	Кръстев	12101114	125	3
12101115	Боян	Кръстев	12101115	112	5
12101115	Боян	Кръстев	12101115	136	5

11

## SQL

**SELECT ... FROM Student, Grade WHERE (Student.ID = Grade.ID)  
AND ...**

ID	First	Last	ID	Code	Mark
12101114	Ангелина	Пеева	12101114	112	4
12101114	Ангелина	Пеева	12101114	125	3
12101115	Боян	Кръстев	12101115	112	5
12101115	Боян	Кръстев	12101115	136	5
12101117	Георги	Пешев	12101117	172	3
12101118	Ангел	Кръстев	12101118	112	6
12101118	Ангел	Кръстев	12101118	125	5
12101118	Ангел	Кръстев	12101118	136	4

12

## SQL

**SELECT ... FROM Student, Grade WHERE (Student.ID = Grade.ID)  
AND (Mark >4)**

ID	First	Last	ID	Code	Mark
12101115	Боян	Кръстев	12101115	112	5
12101115	Боян	Кръстев	12101115	136	5
12101118	Ангел	Кръстев	12101118	112	6
12101118	Ангел	Кръстев	12101118	125	5

13

## SQL

**SELECT First, Last, Mark FROM Student, Grade  
WHERE (Student.ID = Grade.ID) AND (Mark > 4)**

First	Last	Mark
Боян	Кръстев	5
Боян	Кръстев	5
Ангел	Кръстев	6
Ангел	Кръстев	5

14

## SQL

При заявка към няколко таблици почти винаги се използва клаузата **WHERE**, за да се намерят записи със съвпадащи стойности!

**SELECT \* FROM**

*Student, Grade, Course*

**WHERE**

*Student.ID = Grade.ID*

**AND**

*Course.Code = Grade.Code*

15

## SQL

Student

Grade

Course

ID	First	Last	ID	Code	Mark	Code	Title
12101114	Ангелина	Пеева	12101114	112	4	112	Информатика
12101114	Ангелина	Пеева	12101114	125	3	125	Математика
12101115	Боян	Кръстев	12101115	112	5	112	Информатика
12101115	Боян	Кръстев	12101115	136	5	136	Физика
12101117	Георги	Пешев	12101117	172	3	172	ПИК 1
12101118	Ангел	Кръстев	12101118	112	6	112	Информатика
12101118	Ангел	Кръстев	12101118	125	5	125	Математика
12101118	Ангел	Кръстев	12101118	136	4	136	Физика

Student.ID = Grade.ID

Course.Code = Grade.Code

16

## SQL

- За заявка към комбинирани таблици може да се използва JOIN
- Типове JOIN:
  - CROSS JOIN
  - INNER JOIN
  - NATURAL JOIN
  - OUTER JOIN
- OUTER JOIN - допълват се с NULL
- A CROSS JOIN B – връща всички двойки редове от A и B
- A NATURAL JOIN B - връща всички двойки редове с общи стойности
- A INNER JOIN B - връща всички двойки редове, удовлетворяващи условието

17

## SQL

## INNER JOIN.

Съединява две таблици по зададена една колона от всяка от тях. Всеки два реда, за които стойностите в тези колони съвпадат, се 'залепят' един за друг.

Като резултат получаваме таблицата от така получените редове.

SELECT \* FROM Table1, Table2  
WHERE Table1.ColName1 = Table2.ColName3

и

SELECT \* FROM Table1 INNER JOIN Table2  
ON Table1.ColName1 = Table2.ColName3

дават един и същ резултат.

18

## SQL

```
SELECT name FROM lectures, disciplines WHERE
lectures.Id=disciplines.Lid AND disciplines.Term=1;
```

```
SELECT name FROM lectures INNER JOIN disciplines ON
lectures.Id=disciplines.Lid WHERE disciplines.Term=1;
```

В първия случай резултатната таблица е декартовото произведение на двете таблици и допълнително налагаме условие над нейните записи.

При втория пример таблицата, над която работим е получена с определено условие. Заявката се извършва върху нея.

19

## SQL

Ако във FROM има **три** таблици - WHERE е с **две** join условия, които свързват последователно колони от трите таблици.

20

## SQL

**Правила при изпълнение на оператор SELECT над няколко таблици :**

1. Формира се произведението на таблиците от фразата FROM.
2. Ако има фраза WHERE, то се проверява условието в нея за всеки ред на произведението и се избират редовете, за които условието е true.
3. За всеки от избраните редове се изчисляват последователно елементите в целевия списък и се формира резултата.
4. Ако има ключова дума DISTINCT, то се изключват повтарящите се редове.
5. Ако има фраза ORDER BY, то се сортират редовете в резултата.
6. Ако има фраза INTO TEMP, то резултатът се съхранява като временна таблица.

21

## SQL

**INNER JOIN** – резултат, съдържащ записи, които имат съответствие и в двете таблици.

Ако за **всички редове** от една таблица трябва да се съпоставят редове от друга таблица - **LEFT OUTER JOIN**

```
SELECT * FROM lectures LEFT OUTER JOIN disciplines
ON lectures.Id=disciplines.Lid ;
```

**LEFT** - всички записи от **лявата** таблица трябва да присъстват в резултата или **RIGHT** - всички записи от **дясната** таблица трябва да присъстват в резултата.

Ако на всеки запис от първата таблица (която сме определили с **LEFT** или **RIGHT**) съответства запис от втората, тогава **OUTER JOIN** ще дава същите резултати като **INNER JOIN**.

22

## SQL

Външното съединение - **OUTER JOIN** е разширение на обикновеното, наричано още вътрешно ( **INNER JOIN**).

**Правилата при изпълнението му за две таблици са следните:**

1. Създава се вътрешното съединение на двете таблици.
2. Всеки несвързан ред от първата таблица се включва в резултата, като на всички колони от втората таблица се присвояват NULL значения.
3. Всеки несвързан ред от втората таблица се включва в резултата, като на всички колони от първата таблица се присвояват NULL значения.

23

## SQL

**OUTER JOIN**

Всъщност съединението, създадено по тези правила се нарича **пълно външно съединение**, за **ляво външно съединение** - стъпки 1 и 2 и за **дясно външно съединение** -стъпки 1 и 3.

Начините за обозначаване на външните съединения са доста различни в различните СУБД, тъй като те се реализират преди да са включени в стандартите.

Първата (лявата) таблица се нарича **главната**, втората - **подчинена** в съединението.

24

## SQL

- **A INNER JOIN B** - връща всички двойки редове, удовлетворяващи условието.
- **OUTER JOIN** - допълва се с **NULL**.
- **A CROSS JOIN B** – връща всички двойки редове от A и B
- **A NATURAL JOIN B** - връща всички двойки редове с общи стойности

25

## SQL

## CROSS JOIN

SELECT \* FROM Student CROSS JOIN Choice

ID	Name	ID	Name	ID	Code
12101114	Ангелина	12101114	Ангелина	12101114	125
12101115	Боян	12101115	Боян	12101114	125
12101116	Венета	12101116	Венета	12101114	125
12101117	Георги	12101117	Георги	12101114	125
12101114	125	12101114	Ангелина	12101115	112
12101115	112	12101115	Боян	12101115	112
12101115	136	12101116	Венета	12101115	112
12101117	172	12101117	Георги	12101115	112
		12101114	...	...	...

26

## SQL

## NATURAL JOIN

SELECT \* FROM Student NATURAL JOIN Choice

ID	Name	ID	Name	ID	Code
12101114	Ангелина	12101114	Ангелина	12101114	125
12101115	Боян	12101115	Боян	12101115	112
12101116	Венета	12101116	Боян	12101115	136
12101117	Георги	12101117	Георги	12101117	172
ID	Code				
12101114	125				
12101115	112				
12101115	136				
12101117	172				

27

## SQL

## CROSS u NATURAL JOIN

1. SELECT \* FROM A CROSS JOIN B

е равностойно на

SELECT \* FROM A, B

2. SELECT \* FROM A NATURAL JOIN B

е равностойно на

SELECT \* FROM A, B WHERE A.col1 = B.col1  
AND A.col2 = B.col2  
AND ...

28

## SQL

## INNER JOIN

- При **INNER JOIN** се задава условие, което двойките редове да удовлетворят

SELECT \* FROM A INNER JOIN B ON &lt;condition&gt;

- Може да се използва и

SELECT \* FROM A INNER JOIN B USING (col1, col2,...)

- Избира редовете за които дадените колони имат съпадащи стойности

29

## SQL

## NATURAL JOIN

SELECT \* FROM Student INNER JOIN Choice USING (ID)

ID	Name	ID	Name	ID	Code
12101114	Ангелина	12101114	Ангелина	12101114	125
12101115	Боян	12101115	Боян	12101115	112
12101116	Венета				
12101117	Георги				
ID	Code				
12101114	125				
12101115	112				
12101115	136				
12101117	172				

30

## SQL

**SELECT \* FROM Buyer INNER JOIN Property ON Price <= Budget**

Buyer

Name	Budget
Петров	100,000
Иванова	150,000
Ляпчева	80,000

Name	Budget	Address	Price
Петров	100,000	Младост 4 бл.405, оф.1	85,000
Иванова	150,000	Младост 4 бл.405, оф.1	85,000
Иванова	150,000	Бизнес парк бл.3, оф.22	125,000

Property

Address	Price
Младост 4 бл.405, оф.1	85,000
Бизнес парк бл.3, оф.22	125,000
Люлин 9 бл.911, оф.2	175,000

31

## SQL

## JOINS и WHERE клаузи

- Може и без клаузата JOIN
- Същият ефект със заявка към много таблици и клаузата WHERE
- Да се използва ли JOIN или не?
- Да, защото
  - Кратки и ясни заявки
  - NATURAL JOIN е широко известен
- Не, защото
  - Поддржката на JOIN варира в различните диалекти на SQL

32

## SQL

## Писане на заявки 1

- При писане на заявки:
  - Често има много начини да се напише една заявка
  - Важното е заявката да бъде вярна, ясна и кратка в този порядък
  - Не е важно да бъде "хитра" или ефективна

33

## SQL

## Писане на заявки 2

- Повечето СУБД имат оптимизатор на заявките (query optimisers)
  - Той взема заявката на потребителя и очертава как ефективно да се изпълни
  - Ясната и кратка заявка – по-лека за оптимизиране
  - Има начини за подобряване ефективността

34

## БАЗИ ДАННИ

## За тази лекция:

Songs				
cID	Num	Name	Sec	aID
1	1	Момиче	336	1
1	2	Слънчев ден	152	1
1	3	Когато	233	1
2	1	Луната спи	128	2

CD		
cID	Name	Price
1	Всеки	10,00
2	Сингъл	12,00

Artist	
aID	Name
1	Младен
2	Нети

доц. Стоянова

35

## БАЗИ ДАННИ

- Намерете списък от заглавия на CD -та...
- Намерете списък от заглавия на изпълнения, които са по-дълги от 300 секунди...
- Намерете диск с цена > 10 лева
- Намерете всички изпълнения на Младен
- Намерете всички изпълнения на Младен, които са по-дълги от 200 сек.
- Песента Момиче от кой диск е
- Намерете песните от диск с име Всеки
- Кой пее песента Когато
- Намерете списък от имената на изпълнители, които имат изпълнения върху диск (CD) с име "Всеки"...

доц. Стоянова

36

