

... табличка с семизнач 3 ...

Группировка:

select "кадр", count(*), level_sem, min(level_sem)
from prof-table
group by "кадр" -- ключи группировки
having count(*) > 3 -- скаляр => это м.б. скалярным запросам

Определим таблицу в виде

кадр	id	name	level_sem	level_e	count	min
ИЗ 7	5	1
ИЗ 4	1	5
ИЗ 12	1	6.3

В having можно использовать агрегатные функции, а в where нельзя

Из-за порядка обработки запросов не следует в having использовать псевдонимы из select

Order by

order by "кадр" [ASC/DESC]
по возр. по убыв.
ключи сортировки по убыв.

В order by можно исп. псевдонимы из select

Соединения (JOIN)

Логические

как мы управляем соединениями объектов

Физические

как СУБД управляет соединениями объектов (nested loops join; for if...)

Внутренние (INNER)

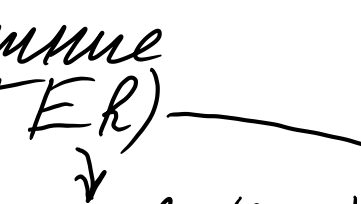


(каж.)

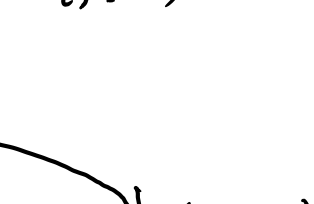
Внешние (OUTER)



(не каж.)



(не каж.)



(каж.)

A не перв. ключ, поэтому

id name_a

1 a

2 b

3 c

4 null

5 null

6 null

7 null

8 null

9 null

10 null

11 null

12 null

13 null

14 null

15 null

16 null

17 null

18 null

19 null

20 null

21 null

22 null

23 null

24 null

25 null

26 null

27 null

28 null

29 null

30 null

31 null

32 null

33 null

34 null

35 null

36 null

37 null

38 null

39 null

40 null

41 null

42 null

43 null

44 null

45 null

46 null

47 null

48 null

49 null

50 null

51 null

52 null

53 null

54 null

55 null

56 null

57 null

58 null

59 null

60 null

61 null

62 null

63 null

64 null

B

id name_b

1 a

2 b

3 c

4 null

5 null

6 null

7 null

8 null

9 null

10 null

11 null

12 null

13 null

14 null

15 null

16 null

17 null

18 null

19 null

20 null

21 null

22 null

23 null

24 null

25 null

26 null

27 null

28 null

29 null

30 null

31 null

32 null

33 null

34 null

35 null

36 null

37 null

38 null

39 null

40 null

41 null

42 null

43 null

44 null

45 null

46 null

47 null

48 null

49 null

50 null

51 null

52 null

53 null

54 null

55 null

56 null

57 null

58 null

59 null

60 null

61 null

62 null

63 null

64 null

select A.*, B.*

from A [INNER] JOIN

on A.id = B.id

часть соединения

(по сути предикат)

null не стыкуется

с null, т.к. null != null

Inner Join

Left Join

Right Join

Full Join

как сделать



select A.*, B.*

from A left join B

on A.id = B.id -- или можно A.id <> B.id

where B.id is null -- это

по логике делаем операцию после join;

Все join'ы с вычитанием - anti join'ы

on A.id = B.id

where B.id is null

or A.id is null

как СУБД выполняет JOIN

3 алгоритма:

1) Nested loops join:

+ работает с любыми условиями

- O(n*m) - квадратичная сложность

2) Merge Join:

данные должны быть предв. отсортированы

и данные у нас были и не отсорти-

рованные => это применяется крайне

редко

+ O < O(n*m)

- сортировка

3) Hash Join

key value

hash(1) { ... }

hash(2) { ... }

...

hash(id) { ... }

+ время

- зап. ресурс (память)

- хранилищ у hash-функций

- в запросе должно быть только =, т.к.

1 < 2 <=> hash(1) < hash(2)

Exist всегда переопределяются

в left anti join

Задачи:

1) id sub S

1 sub1

2 sub2

id bonus B

2 bonus1

3 bonus2

id sub bonus

1 sub1 null

2 sub2 bonus1

3 null bonus2

Syncrpt Full join:

select coalesce(S.id, B.id)

from S Full join B

on S.id = B.id

серия первое

не нулевое

значение

8. sub

8. bonus

значение

parameter

id p-name p-value

1 name Julia

gender F

2 name Ivan

id name gender

1 Julia F

2 Ivan null