

как из массива выбрать все записи с теми же ID? А с разными?

select * from table where id % 2 = 0 positive

select * from table where id % 2 != 0 negative

Важен ли в составе SQL идексе порядок столбцов?

Важен. Когда строится составной индекс, всегда используется первая стобец, а номера следующих индексов определяются на основе первого.

Можно ли сделать образец заголовка зарезервное слово AUTO_INCREMENT? Да.

set identity-insert tables on

insert into table1 (id, name)

select id, name from temptbl

set identity-insert off

Чтобы удалить из массива A:

select sum(case when x > 0 then x else 0 end)

sum-pos,

sum(case when x < 0 then x else 0 end)

sum-neg

from A

Как скопировать записи из tab1 в tab2?

Insert into tab2 (col1, col2, col3, ...)

Select col1, col2, col3, ...

From tab1 Where condition

SQL Inner Join

Select c.FirstName, c.LastName, p.ProductName,
Sum(sd.Quantity * p.ProductPrice) as TotalPrice
From Customer c join Sale s On
c.CustomerId = s.CustomerId
join SaleDetail sd On
s.SaleId = sd.SaleId
join Product p On
p.ProductId = sd.ProductId
Where s.OrderDate Between '3/1/2021' and '3/31/2021'
Group By c.FirstName, c.LastName, p.ProductName

Kak nowoznacze nowego rekordu Id
Jezu uchowaz obecne id i dodaj max?
select top 1 id from table order by id desc

Ten oznaczaemce IN om EXISTS?

IN

Albion Cos 3218 *
Mujer surfice si los *
Almeria Surfice Gato allen *
Alkyl
Alonj P1313 88 *
Alkyl of 321282
Alfiz dojer surfice

EXISTS

Albion Surfice P13218 *
Mujer Surfice Prever *
Almeria Surfice Alkyl 88 *
Alonj dojer Surfice 321282
Alfiz

Where vs Having

Select c.Id-Client, s.Date-Sale, s.Sum-Payment

From Clients c

Join Sales s

On c.Id-Client = s.Id-Client

Where s.Date-Sale Between

to_date('01.01.2019', 'dd.mm.yyyy') And
to_date('31.12.2019', 'dd.mm.yyyy')

Select c.Id-Client, Sum(s.Sum-Payment) As Total-Sum

From Clients c

Join Sales s

On c.Id-Client = s.Id-Client

Where s.Date-Sale Between

to_date('01.01.2019', 'dd.mm.yyyy') And
to_date('31.12.2019', 'dd.mm.yyyy')

Group By c.Id-Client

Having Sum(s.Sum-Payment) > 100000

Что такое Index?

Индекс относится к методу инсталляции производительности, позволяющему быстрее извлекать записи из таблицы. Индекс создает отдельную структуру для индексируемого поля и агрегатов, позволяя быстрее получать записи.

Unique Index: — не позволяет иметь несколько повторяющихся значений, если столбец индексируется уникально. Если первичный ключ определяет, уникальный индекс может быть применен автоматически.

Clustered Index: — изменяет физический порядок таблицы и выполняет поиск на основе значений ключа. Каждая таблица может иметь только один мастеризованный индекс.

Non-Clustered Index: — не изменяет физический порядок таблицы и поддерживает логический порядок записей. Каждая таблица может иметь много некластеризованных индексов.

Что такое ACID в базе данных?

ACID - означает Atomicity (атомарность)
Consistency (согласованность)
Isolation (изолированность)
Durability (долговечность)

Он используется для обеспечения надежной обработки транзакций данных в системе базы данных

<< Триgger >> в SQL

Триgger - особенность языка хранимых процедур, которая предполагает что автоматическое выполнение в момент или после изменения данных. Это позволяет вам выполнить некое кода, когда вставка, обновление или удаление какой запрос выполняется к определенной таблице.

В SQL доступны 3 оператора.

1. Арифметические
2. Логические
3. Операторы сравнения

Сравнением не значение Null с 0 или ""
Значение Null не равно 0 или пробелу.
Значение Null представляет значение, которое недоступно, недействительно, присвоено или неприменимо как 0 - это число, а пробел - символ.

Покажите мне, все. заплаты у продавца где
без премий (岗位 sales man), и которых
все. заплата (sal) превосходит (comm). sort sal desc
select ename, sal, comm from table "ename"
where sal > nvl(comm, 0) and job = 'salesman'
order by sal desc, ename

Напишите чеки для которых, вкладывают
их мне в оклад. Если оклад ниже \$1500, то
запишите это на сообщение "Below 1500",
Значение \$1500 замените на "On Target"

Select ename

case

when sal < 1500 then 'Below 1500'

when sal = 1500 then 'On Target'

else null

end as salary

From table

Table `tbl` nowe nmbr

1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1

0 → 2, 1 → 3

update `tbl` set `nmbr` = case when `nmbr` = 0
then 2 else 3 end

Некоторые SQL-запросы, с применением
`UNION ALL` (не `UNION`), использующие
where где имеющие губинскатов

`select * from table where a=X`

`UNION ALL`

`select * from table where b=Y and a!=X`

`VARCHAR` vs `NVARCHAR` vs `CHAR`

`NVARCHAR` - может хранить Unicode - 1 символ 2 байта
кодировки занимает 65536 символов

`VARCHAR` - хранение в формате ASCII
каждый символ занимает 1 байт
кодировки занимает 256 символов

~~`CHAR`~~ `varchar(10)` - может хранить до 10
символов и не может хранить символы
которые нужно указать

например 6, 8 или 2

`CHAR` - `char(10)` - может хранить только 10
символов и не может хранить символы
которые нужно указать

Delete vs Truncate

Delete

- * Используется для удаления строк в таблице
- * Вы можете восстановить данные после удаления
- * DML - команда
- * Медленнее, чем Truncate

Truncate

- * — удаление всех строк
- * Вы не можете восстановить данные
- * DDL - команда
- * Быстрее

DDL - позволяет выполнять различные операции с базой данных, CREATE, ALTER, DROP

DML - позволяет получать доступ к данным и манипулировать ими, например, вставлять, обновлять, удалять, извлекать из базы данных (Insert, Update, Delete, Select)

DCL - позволяет контролировать доступ к базе данных. Пример - GRANT (предоставление прав) REVOKE (отзывание прав)

Что такое ограничение (Constraints)?

- | | |
|------------|---------------|
| * Not Null | * Primary key |
| * Check | * Foreign key |
| * Default | |
| * Unique | |

Ограничение используется для указания ограничения на тип данных таблицы. Они могут быть указаны при создании или изменении таблицы.