

Лабораторийн ажил 8. ӨС-гийн боловсруулалт-Сторед функц, View-Виртуал хүснэгт

Лабораторийн ажлын зорилго

- сторед функц бичиж сурах
- сторед функц руу параметр дамжуулах, буцах утгыг хэрэглэх
- Өгөгдөл боловсруулахад хувьсагч, давталт, нөхцөл шалгах үйлдлүүд ашиглах
- View-Виртуал хүснэгт байгуулж сурах
- View-Виртуал хүснэгт дотор хувьсагч хэрэглэх

Даалгавар-1

Оюутан бүр Бие даалтын ажлаар сонгосон сэдвийн дагуу байгуулсан ӨС схемийг ашиглаж ӨС үүсгэх скрипыг ажиллуулж , ӨС-г шинээр үүсгэх.

- Бие Даалт №1,2 дээр үүсгэсэн (Лавлах хүснэгтүүдэд бүх бичлэгийг оруулсан байх - Бодит утгууд оруулна) скрипт файлыг хуулж Лаборатори №8 файл үүсгэ
- Лаборатори №8 скрипт файлыг ажиллуулж ӨС-г үүсгэ

Даалгавар-2

Оюутан бүр Бие даалтын ажлаар сонгосон сэдвийн дагуу байгуулсан Функциональ шаардагад хэрэглэгдэх **тооцоолох/боловсруулах** үйлдлийг хийх Сторед функц үүсгэх.

- Оролтын параметр, буцаах утга бүхий Сторед функц үүсгэх
- Шаардлагтай тооцооллыг хийж үр дүнг буцаах

Даалгавар-3

БД-н Функциональ шаардлага дээр тодорхойлсон жагсаалт, хайх хийх **ОБЪЕКТЫН ЖАГСААЛТ ХАРАХ, ОБЪЕКТЫГ ХАЙХ(Params), _____ТАЙЛАН ГАРГАХ(Params)** функцүүдийн Кверийг **View-Виртуал хүснэгт** ашиглаж бүтээх

- Энгийн View-Виртуал хүснэгт үүсгэх
- Холбоос бүхий View-Виртуал хүснэгт үүсгэх /олон хүснэгт холбосон-join/
- Хувьсагч ашигласан View-Виртуал хүснэгт үүсгэх
- Оюутан тус бүр дор хаяж НЭГ ширхэг **ОБЪЕКТЫН View-Виртуал хүснэгтүүдийг бүтээж лабораторийг тооцно**
- Баг тус бүр дор хаяж **ХОЁР ширхэг ОБЪЕКТЫН View-Виртуал хүснэгтүүдийг бүтээж БД-ыг тооцно**

Хугацаа

Лабораторийн цаг дээр үзүүлнэ.

Гүйцэтгэж хамгаалах шаардлага

- Лабораторийн ажил №8 скрипт файлаас ӨС-ийн үүсгэх
- Даалгавар №2,3 -оор сторед функц бүтээсэн кверийг тайлбарлах
- Бүтээсэн сторед функцийг ажиллуулж, тайлбарлах

Нэмэлт материал

Онолын хэсэг, жишээ

Функц

SQL болон T-SQL хэлэнд байдаг зарим функцүүдтай танилцсан:

- ROUND
- AVG
- SUM
- COUNT гм...

Мөн бид өөрсдөө **Сторед функц** эсвэл **UDF** үүсгэж ашиглах боломжтой.

Сторед функц

- Сторед процедуртай адил сервер дээр хадагдаж, түүнийг дурын байрлал дуудаж ашигладаг.

Stored Function vs SQL Statement

SQL Statement

First Time

- Check syntax
- Compile
- Execute
- Return data

Second Time

- Check syntax
- Compile
- Execute
- Return data

Stored Function

Creating

- Check syntax
- Compile

First Time

- Execute
- Return data

Second Time

- Execute
- Return data

Сторед функц ба сторед процедура

- Функц нь сторед процедурыг бодвол арай цөөн боломжтой.
- Функц нь зөвхөн нэг утга буцаана.
- Функцийг SELECT илэрхийлэл дотор дуудаж ашиглаж болдогоороо сторед процедураас ялгаатай.

Сторед функцийн бүтэц

- **RETURN** түлхүүр үг нь заавал байх ёстой бөгөөд энэ нь функцээс буцах утгын төрлийг зааж өгдөг.

Синтакс:

```
DELIMITER //
```

```
CREATE FUNCTION Функцийн_нэр (param)
```

```
    RETURNS Datatype
```

```
    BEGIN
```

```
        --code
```

```
        RETURN var
```

```
    END //
```

```
DELIMITER ;
```

Сторед функцийн хэрэглээ- Тооцоолол

Сторед функц дотор дурын тооцоолол хийх боломжтой. Үүнд:

- Хувьсагч үүсгэх, ашиглах
- Нөхцөл шалгах
- Давталт хийх
- Арифметик, логик тооцоолол хийх
- T-SQL ажиллуулах
- Функц дотор процедур болон функц дуудах

Сторед функцийн хэрэглээ

Курсор-Тооцоолол

- Курсор нь query-ний үр дүнд үүссэн бичлэгийн олонлогийн бичлэг тус бүрт давталт ашиглан хандаж тодорхой үйлдлийг гүйцэтгэх боломжийг олгодог.
- Үйлдэл гүйцэтгэх хэсгийг Сторед функц ашиглаж ихэвчлэн тооцоолол хийдэг.

	buy_id	buy_book_id	buy_qty	buy_ord_id	bok_id	bok_name	bok_price	bok_status	
	1	1	2	1	1	Twilight	100.00	1	→ Үйлдэл гүйцэтгэх
	3	2	2	1	2	Davinci Code	10.00	2	→ Үйлдэл гүйцэтгэх
	4	1	1	1	1	Twilight	100.00	1	→ Үйлдэл гүйцэтгэх
	5	2	1	1	2	Davinci Code	10.00	2	→ Үйлдэл гүйцэтгэх

Сторед функц: Жишээ

- Нийт дүнг боддог функц бичье

```

DELIMITER //

CREATE FUNCTION fc_tot_qty

(   price INT,
    quantity INT
) RETURNS INT

BEGIN
    DECLARE tot_qty INT;

    SET tot_qty = price*quantity;

    RETURN tot_qty;

END //

DELIMITER ;

```

Оролтын параметр

Функцийн буцаах утгын төрөл

Функцийн Эхлэл, төгсгөл

Функцийн боловсруулалт хийх хэсэг

Функцийн буцаах утга

Функц үүсгэх эрх

Функц үүсгэх Эрх олгох:

```
use ӨСнэр;
```

```
SET GLOBAL log_bin_trust_function_creators = 1;
```

Default буюу анхны утга 0 байдаг тул функц үүсгэх командыг дуудахаас өмнө эрхийг сольсон байх хэрэгтэй. Үгүй бол алдаа мэдээлэл гарж үүсгэхийг хориглоно.

```
Error Code: 1418. This function has none of DETERMINISTIC, NO  
SQL, or READS SQL DATA in its declaration and binary logging  
is enabled (you *might* want to use the less safe  
log_bin_trust_function_creators variable)
```

Функцийг дуудах :

Синтакс:

```
SELECT func_name (param) ;
```

Үүсгэсэн функцээ илүү олон үйлдэл бүхий SELECT илэрхийлэлд ч ашиглах боломжтой!

Сторед функц: Жишээ:

Str –тэй ажилладаг функц:

```
CREATE FUNCTION fc_welcome( str CHAR(20))  
    RETURNS CHAR(50)  
    RETURN CONCAT('WELCOME TO ',str,'!');
```

Функцийг дуудах:

```
SELECT fc_welcome('UB');
```

Үр дүн:

```
WELCOME TO UB!
```

	fc_welcome('UB')
▶	WELCOME TO UB!

Сторед функц: Жишээ:

Өгөгдсөн номны нийт үнийг боддог функц:

```
DELIMITER //  
CREATE FUNCTION fc_tot_qty1(bokId INT)  
    RETURNS INT  
BEGIN  
    DECLARE tot_qty INT;  
    SELECT SUM(bok_price*buy_qty) INTO tot_qty  
        FROM book, buy  
        WHERE bok_id=buy_bok_id
```

```
        AND bok_id=bokId;  
    RETURN tot_qty;  
END //  
DELIMITER ;
```

Функцийг дуудах:

```
SELECT fc_tot_qty1(1);
```

Үр дүн: 200

bok_price	buy_qty
100	1
100	1

Сторед функц: Жишээ:

INSERT/UPDATE/DELETE үйлдүүдэд функц ашиглах бол RETURN 1 буцах утгыг ашиглана:

```
DELIMITER //  
CREATE FUNCTION fc_insert  
(    bokName VARCHAR(255),  
        bokPrice INT  
)    RETURNS INT  
BEGIN  
    INSERT INTO book VALUES (NULL,bokName,bokPrice);  
    RETURN 1;  
END //  
DELIMITER ;
```

Функцийг дуудах:

```
SELECT fc_insert('name',1500);
```

Үр дүн:

bok_id	bok_name	bok_price
1	Twilight	100
2	Da Vinci Code	10
3	Very complica...	20
4	strange book	1000
5	name	1500

Функцийг устгах синтакс:

```
DROP FUNCTION [IF EXISTS]    Функцийн_нэр;
```

View:

- SQL View гэдэг нь SELECT илэрхийлэлд суурилсан виртуал хүснэгт юм.
- View нь ерөнхийдөө ӨС-ийн энгийн хүснэгттэй ижил боловч гол ялгаа нь энгийн хүснэгт бодит өгөгдлийг агуулдаг бол view нь бодит өгөгдөл агуулдаггүй.
- View-ийн өгөгдөл нь view-г дуудах үед автоматаар үүсгэгдэнэ.
- View нь ерөнхийдөө SELECT илэрхийллийг дахин ашиглах боломжтой виртуал хүснэгт рүү багцлах үйлдэл юм.
- View нь нэг болон хэд хэдэн хүснэгт эсвэл view-тэй холбогдоно.

View үүсгэх:

Синтакс:

```
CREATE VIEW view_нэр  
        AS select илэрхийлэл;;
```

view нь дараах зүйлүүдтэй холбогдож, эсвэл ашиглагдаж болно:

- Бусад view,
- SQL query,
- Сторед процедур,
- Функц

VIEW-г ашиглахдаа:

Синтакс:

```
SELECT * FROM view_нэр;
```

View : Жишээ

```
CREATE VIEW v_product AS  
SELECT * FROM product;
```

Зарлах хэсэг

Бичлэг жагсааж
гаргах SELECT
командын хэсэг

• Дуудах:

```
SELECT * FROM v_product;
```

prd_id	prd_name
1	tea
2	milk

• view нэрлэх: v_нэр

VIEW Жишээ:

Синтакс:

```
CREATE VIEW v_book AS
SELECT *
FROM book, buy, orders
WHERE bok_id= buy_bok_id
AND buy_ord_id = ord_id;
```

View-г дуудах:

```
SELECT * FROM v_book;
```

bok_id	bok_name	bok_price	buy_id	buy_ord_id	buy_bok_id	buy_month	buy_qty	ord_id	ord_cus_id
1	Twilight	100	1	1	1	10	1	1	1
2	Da Vinci Code	10	2	2	2	10	2	2	2
2	Da Vinci Code	10	3	1	2	11	2	1	1
1	Twilight	100	4	1	1	11	1	1	1
2	Da Vinci Code	10	5	1	2	11	2	1	1

VIEW-г устгах:

Синтакс:

```
DROP VIEW view_нэр;
```

Жишээ:

```
DROP VIEW v_book;
```

View дотор хувьсагч ашиглах:

- View нь процедур, функц шиг бүтэцгүй тул параметр дамжуулах боломжгүй
- Харин view дотор глобал хувьсагч ашиглаж болдог
- Олон хүснэгтийг холбож ашиглах боломжтой

```
DELIMITER //
```

```
Create function fc_oid() returns int
```

```
BEGIN
```

```
    if @oid is null Then SET @oid=0;
```

```
    end if;
```

```
    return @oid;
```

```
END //
```

```
DELIMITER ;
```

Функцыг дуудах:

```
SET @oid=1;
```

```
SELECT fc_oid();
```

View дотор хувьсагч ашиглах

- Захиалгын дэлгэрэнгүй **хайх(Захиалгы№)**
- ```
SELECT orders.ord_id,book.bok_name,
 book.bok_price, buy.buy_qty, orders.ord_date
FROM orders , buy, book

where orders.ord_id=buy.buy_bok_id
and buy.buy_bok_id=book.bok_id
and orders.ord_id=1;
```

|   | ord_id | bok_name | bok_price | buy_qty | ord_date   |
|---|--------|----------|-----------|---------|------------|
| ▶ | 1      | Twilight | 100.00    | 2       | 2012-04-10 |
|   | 1      | Twilight | 100.00    | 1       | 2012-04-10 |

## View дотор хувьсагч ашиглах

- Захиалгын дэлгэрэнгүй **хайх(Захиалгы№)**
- ```
CREATE VIEW v_book_find_id AS  
  
SELECT orders.ord_id, book.bok_name,  
       book.bok_price, buy.buy_qty, orders.ord_date  
FROM orders , buy, book  
  
where orders.ord_id=buy.buy_bok_id  
and   buy.buy_bok_id=book.bok_id  
  
and orders.ord_id=fc_oid();
```

View дотор хувьсагч ашиглах

- Захиалгын дэлгэрэнгүй **хайх(Захиалгы№)**
- ```
set @oid=1;
```
- ```
select * from v_book_find_id;
```

	ord_id	bok_name	bok_price	buy_qty	ord_date
▶	1	Twilight	100.00	2	2012-04-10
	1	Twilight	100.00	1	2012-04-10

- ```
set @oid=2;
```
- ```
select * from v_book_find_id;
```

	ord_id	bok_name	bok_price	buy_qty	ord_date
▶	2	Davinci Code	10.00	1	2012-04-09
	2	Davinci Code	10.00	2	2012-04-09
	2	Davinci Code	10.00	1	2012-04-09