

## Лабораторийн ажил 5. INNER, LEFT, RIGHT холболт, SUBQUERY

### Даалгавар-1

Оюутан бүр Бие даалтын ажлаар сонгосон сэдвийн дагуу байгуулсан ӨС схемийг ашиглаж ,  
файлаа үргэлжлүүлж Функционал шаардлага дээр тодорхойлсон мэдээлэл шүүж, жагсааж  
гаргах хэсгийн Квери/query бүтээж даалгавараа гүйцэтгэнэ

- Бие Даалт №1 дээр үүсгэсэн (Лавлах хүснэгтүүдэд бүх бичлэгийг оруулсан байх -Бодит  
утгууд оруулна) скрипт файлыг хуулж Лаборатори №5 файл үүсгэ
- Лаборатори №5 скрипт файлыг ажиллуулж ӨС-г үүсгэ

### Даалгавар-2

БД-н Функционал шаардлага дээр тодорхойлсон мэдээлэл шүүж, жагсааж гаргах хэсгийн  
Кверийг **INNER JOIN, LEFT JOIN, RIGHT JOIN** ашиглаж бүтээх

- Хоёр түүнээс олон хүснэгтийг / INNER JOIN, LEFT JOIN, RIGHT JOIN / холбож мэдээллийг  
харуулах
- INNER JOIN, LEFT JOIN, RIGHT JOIN – **Тус бүр дор хаяж нэг ширхэг квери бүтээнэ**

### Даалгавар-3

БД-н Функционал шаардлага дээр тодорхойлсон мэдээлэл шүүж гаргах хэсгийн Кверийг **WHERE  
хэсэгт SUB QUERY** ашиглаж бүтээх

- Мэдээллийг шүүж харуулах хэсэгт **SUB QUERY** ашиглах
- Ж:Менежер төрөлтэй ажилчдыг харах  
`select id,name,phone, address from Employee`  
`WHERE emp_type_id=(select t_id from Employee_type where type='Manager')`
- Дор хаяж хоёр ширхэг квери бүтээнэ

### Даалгавар-4

БД-н Функционал шаардлага дээр тодорхойлсон мэдээлэл шүүж, жагсааж гаргах хэсгийн  
Кверийг **Холбох /JOIN/ хэсэгт SUB QUERY** ашиглаж бүтээх

- Холболт /JOIN / хийх хэсэгт ХҮСНЭГТ\_ЭХ эсвэл ХҮСНЭГТ\_ОХИН хүснэгтийг SUB QUERY  
ашиглаж бүтээх
- Ж:Менежер төрөлтэй ажилчдыг харах  
`select e.id, e.name, e.phone, e.address, t.type from Employee as e JOIN`  
`(select t_id, type from Employee_type where type='Manager') as t`  
`on e.emp_type_id=t.t_id`
- Дор хаяж хоёр ширхэг квери бүтээнэ

### Хугацаа

Лабораторийн цаг дээр үзүүлнэ.

## Гүйцэтгэж хамгаалах шаардлага

- Лабораторийн ажил №5 скрипт файлаас ӨС-ийн үүсгэх, өгөгдөл нэмэх хэсгийг ажиллуулна
- Даалгавар №1 INNER JOIN, LEFT JOIN, RIGHT JOIN ашиглаж бүтээсэн кверийг ажиллуулж тайлбарлах
- Даалгавар №2 Мэдээлэл шүүж WHERE хэсэгт SUB QUERY ашиглаж бүтээсэн кверийг ажиллуулж тайлбарлах
- Даалгавар №2 Холбох /JOIN/ хэсэгт SUB QUERY ашиглаж бүтээсэн кверийг ажиллуулж тайлбарлах

## Нэмэлт материал

### Онолын хэсэг, жишээ

Жишээнд ашиглагдах **Genders** болон **Persons** хүснэгтүүдийг үүсгэе.

```
CREATE TABLE Genders (  
    GenderID INT NOT NULL AUTO_INCREMENT,  
    Gender VARCHAR(15),  
    CONSTRAINT PK_Genders PRIMARY KEY (GenderID)  
);
```

```
INSERT INTO Genders(Gender)  
VALUES('Male'),('Female'),('Unknown');
```

```
CREATE TABLE Persons  
(  
    PersonID int not null auto_increment,  
    FirstName varchar(20),  
    LastName varchar(20),  
    GenderID int,  
    CONSTRAINT PK_Persons PRIMARY KEY(PersonID)  
);  
Үүсгэсэн хүснэгтүүддээ өгөгдөл оруулна.
```

#### Жишээ1: INNER JOIN.

Inner join холболтын үр дүнд эх болон охин харьцааны түлхүүрийн утга тэнцүү байх бүх бичлэгүүд гарна.

Хэрэв аль нэг хүснэгтийн утга хоосон /NULL/ байвал үр дүнгийн хүснэгтэнд гарахгүй.

```
SELECT Persons.PersonID, Persons.FirstName,  
        Persons.LastName, Genders.GenderID, Genders.Gender  
FROM Persons INNER JOIN Genders  
ON Persons.GenderID = Genders.GenderID;
```

### Үр дүнгийн хүснэгт:

|   | PersonID | FirstName | LastName | GenderID | Gender  |
|---|----------|-----------|----------|----------|---------|
| ▶ | 1        | John      | Franks   | 1        | Male    |
|   | 2        | Peter     | Sonnens  | 1        | Male    |
|   | 4        | Mary      | Shambers | 2        | Female  |
|   | 5        | Chryssa   | Lurie    | 2        | Female  |
|   | 6        | Hellah    | Zanogh   | 3        | Unknown |
|   | 7        | Olympia   | Sumners  | 2        | Female  |
|   | 8        | Roberta   | Jerseys  | 2        | Female  |

Бүх багана болон түлхүүрийн утгуудыг гаргаж үзье:

```
SELECT * FROM Persons INNER JOIN Genders
ON Persons.GenderID = Genders.GenderID;
```

|   | PersonID | FirstName | LastName | GenderID | GenderID | Gender  |
|---|----------|-----------|----------|----------|----------|---------|
| ▶ | 1        | John      | Franks   | 1        | 1        | Male    |
|   | 2        | Peter     | Sonnens  | 1        | 1        | Male    |
|   | 4        | Mary      | Shambers | 2        | 2        | Female  |
|   | 5        | Chryssa   | Lurie    | 2        | 2        | Female  |
|   | 6        | Hellah    | Zanogh   | 3        | 3        | Unknown |
|   | 7        | Olympia   | Sumners  | 2        | 2        | Female  |
|   | 8        | Roberta   | Jerseys  | 2        | 2        | Female  |

### Жишээ 2. LEFT OUTER JOIN.

Left outer join холболт нь Гадаад түлхүүрээр холбоостой хүснэгтийн зүүн хүснэгт /JOIN - ны зүүн талд байрлах- хүснэгт№1/ бүх болон түлхүүрийн утга тэнцүү холбоостой бичлэгүүдийг харуулах ба гадаад түлхүүр нь утгагүй хоосон байгаа бичлэгүүд NULL утгатай болно.

```
SELECT Persons.PersonID, Persons.FirstName,
Persons.LastName, Genders.GenderID, Genders.Gender
FROM Persons LEFT OUTER JOIN Genders
ON Persons.GenderID = Genders.GenderID;
```

### Үр дүнгийн хүснэгт:

|   | PersonID | FirstName | LastName | GenderID | Gender  |
|---|----------|-----------|----------|----------|---------|
| ▶ | 1        | John      | Franks   | 1        | Male    |
|   | 2        | Peter     | Sonnens  | 1        | Male    |
|   | 3        | Leslie    | Aronson  | NULL     | NULL    |
|   | 4        | Mary      | Shambers | 2        | Female  |
|   | 5        | Chryssa   | Lurie    | 2        | Female  |
|   | 6        | Hellah    | Zanogh   | 3        | Unknown |
|   | 7        | Olympia   | Sumners  | 2        | Female  |
|   | 8        | Roberta   | Jerseys  | 2        | Female  |
|   | 9        | Helene    | Campo    | NULL     | NULL    |

### Жишээ 3: RIGHT OUTER JOIN.

Right outer join холболт нь нь Гадаад түлхүүрээр холбоостой хүснэгтийн баруун хүснэгтийн /JOIN -ны баруун талд байрлах- хүснэгт№2/ бүх болон түлхүүрийн утга тэнцүү холбоостой бичлэгүүдийг харуулах ба гадаад түлхүүр нь утгагүй хоосон байгаа бичлэгүүд NULL утгатай болно.

```
SELECT Persons.PersonID, Persons.FirstName,
       Persons.LastName, Genders.GenderID, Genders.Gender
FROM Persons RIGHT JOIN Genders
       ON Persons.GenderID = Genders.GenderID;
```

#### Үр дүнгийн хүснэгт:

Хүснэгт 2ш мөр бичлэгтэй гэж үзвэл

insert into Persons Values(null,'John', 'Franks', 1);

insert into Persons Values(null,'Peter', 'Sonnens', 1);

|   | PersonID | FirstName | LastName | GenderID | Gender  |
|---|----------|-----------|----------|----------|---------|
| ▶ | 2        | Peter     | Sonnens  | 1        | Male    |
|   | 1        | John      | Franks   | 1        | Male    |
|   | NULL     | NULL      | NULL     | 2        | Female  |
|   | NULL     | NULL      | NULL     | 3        | Unknown |

### Жишээ 4: CSROSS JOIN.

Холболт хийгдсэн 2 хүснэгтийн хоёулангийнх нь бүх бичлэгийг харуулах ба эх харьцааны бичлэг тус бүр нь охин харьцааны хэдэн бичлэгтэй холбогдож байгааг харуулна.

Энэ холболтын үед 2 харьцаанд хоёуланд нь ижил бичлэг байх шаардлагагүй.

```
SELECT Persons.PersonID, Persons.FirstName,
       Persons.LastName, Genders.GenderID, Genders.Gender
FROM Persons CROSS JOIN Genders
```

#### Үр дүнгийн хүснэгт:

Хүснэгтүүд 2ш, 3ш мөр бичлэгтэй гэж үзвэл-Жишээ 3тай ижил

|   | PersonID | FirstName | LastName | GenderID | Gender  |
|---|----------|-----------|----------|----------|---------|
| ▶ | 2        | Peter     | Sonnens  | 1        | Male    |
|   | 1        | John      | Franks   | 1        | Male    |
|   | 2        | Peter     | Sonnens  | 2        | Female  |
|   | 1        | John      | Franks   | 2        | Female  |
|   | 2        | Peter     | Sonnens  | 3        | Unknown |
|   | 1        | John      | Franks   | 3        | Unknown |

## Жишээ 5: Self join

Нэг хүснэгт өөрийг нь өөртэй нь холбож ашиглах хэлбэр юм. Хүснэгтийг өөртэй нь холбохын тулд уг хүснэгтдээ 2 өөр alias нэр өгнө.

Ажилтан бүрийн даргын нэрийг харъя:

Өгөгдсөн хүснэгт

|                       |  |
|-----------------------|--|
| <b>employee</b>       |  |
| emp_id INT            |  |
| emp_name VARCHAR(255) |  |
| emp_report_to INT     |  |
| Indexes               |  |

  

| emp_id | emp_name | emp_report_to |
|--------|----------|---------------|
| 5      | boss1    | NULL          |
| 6      | emp1     | 5             |
| 7      | emp2     | 5             |
| 8      | boss2    | NULL          |
| 9      | emp3     | 8             |

```
SELECT emp.emp_name , boss.emp_name AS boss
      FROM employee boss INNER JOIN employee emp
      ON boss.emp_id = emp.emp_report_to;
```

Үр дүнгийн хүснэгт:

| emp_name | boss  |
|----------|-------|
| emp1     | boss1 |
| emp2     | boss1 |
| emp3     | boss2 |

**Жишээ: per\_id=3 байх хүнтэй ижил улсад амьдардаг хүмүүсийн мэдээллийг харах**

2 аргаар гаргаж болно:

- Subquery
- Self Join

Person хүснэгт:

| per_id | per_first_name | per_last_name | per_dob    | per_pert_id | per_ctr_id |
|--------|----------------|---------------|------------|-------------|------------|
| 1      | first1         | last1         | NULL       | 1           | 1          |
| 2      | first2         | last_name2    | 1980-01-01 | 1           | 2          |
| 3      | first3         | last_name3    | NULL       | NULL        | 1          |

self join ашиглан хялбарчлах:

```
SELECT * FROM person p1 INNER JOIN person p2
      ON p1.per_ctr_id = p2.per_ctr_id
      WHERE p2.per_id= 3
```

### Үр дүнгийн хүснэгт:

| per_id | per_first_name | per_last_name | per_dob | per_pert_id | per_ctr_id | per_id | per_first_name | per_last_name | per_dob | per_pert_id |
|--------|----------------|---------------|---------|-------------|------------|--------|----------------|---------------|---------|-------------|
| 1      | first1         | last1         | NULL    | 1           | 1          | 3      | first3         | last_name3    | NULL    | NULL        |
| 3      | first3         | last_name3    | NULL    | NULL        | 1          | 3      | first3         | last_name3    | NULL    | NULL        |

Sub query: ашиглан хялбарчлах:

```
SELECT * FROM person
WHERE per_ctr_id IN
(SELECT per_ctr_id FROM person WHERE per_id=3);
```

### Үр дүнгийн хүснэгт:

| per_id | per_first_name | per_last_name | per_dob | per_pert_id | per_ctr_id |
|--------|----------------|---------------|---------|-------------|------------|
| 1      | first1         | last1         | NULL    | 1           | 1          |
| 3      | first3         | last_name3    | NULL    | NULL        | 1          |

## Subquery ашиглах

Өгөгдсөн хүснэгт

ORDERS table

|   | emp_id | emp_name |
|---|--------|----------|
| ▶ | 1      | emp1     |
|   | 2      | emp2     |
|   | 3      | emp3     |
|   | 4      | last1    |
|   | 5      | last2    |
| * | NULL   | NULL     |

Employee table

|   | emp_id | emp_name | emp_gender | emp_address | emp_phone |
|---|--------|----------|------------|-------------|-----------|
| ▶ | 1      | emp1     | 1          | mgl ub      | 9999      |
|   | 2      | emp2     | 2          | mgl ub      | 8888      |
|   | 3      | emp3     | 1          | mgl ub      | 7777      |
|   | 4      | last1    | 1          | mgl ub      | 8888      |
|   | 5      | last2    | 2          | mgl ub      | 8888      |

Person table

|   | per_id | per_first_name | per_last_name | per_dob    |
|---|--------|----------------|---------------|------------|
| ▶ | 1      | first1         | last1         | NULL       |
|   | 2      | first2         | last2         | 1980-01-01 |
|   | 3      | first1         | last1         | NULL       |
|   | 4      | first2         | last2         | 1980-01-01 |
|   | 5      | first2         | last2         | 1980-01-01 |
|   | 6      | first2         | last2         | 1980-01-01 |
|   | 7      | first2         | last2         | 1980-01-01 |
|   | 8      | first2         | last2         | NULL       |

Buy table

|   | buy_id | buy_bok_id | buy_qty | buy_ord_id |
|---|--------|------------|---------|------------|
| ▶ | 1      | 1          | 2       | 1          |
|   | 2      | 2          | 1       | 2          |
|   | 3      | 2          | 2       | 1          |
|   | 4      | 1          | 1       | 1          |
|   | 5      | 2          | 1       | 1          |

### Жишээ 5:

Person хүснэгт дэх per\_last\_name-тэй ижил нэртэй хүмүүсийг  
Employee хүснэгтээс харуулах:

```
SELECT * FROM employee
      WHERE emp_name IN
            (SELECT per_last_name FROM person);
```

### Үр дүнгийн хүснэгт:

|   | emp_id | emp_name | emp_gender | emp_address | emp_phone |
|---|--------|----------|------------|-------------|-----------|
| ▶ | 4      | last1    | 1          | mgl ub      | 8888      |
|   | 5      | last2    | 2          | mgl ub      | 8888      |
| * | NULL   | NULL     | NULL       | NULL        | NULL      |

### Жишээ 6:

Хамгийн их тоогоор захиалсан захиалгыг харах:

```
SELECT *FROM orders
      WHERE ord_id =
            (SELECT buy_ord_id FROM buy ORDER BY buy_qty DESC LIMIT 1);
```

### Үр дүнгийн хүснэгт:

|   | ord_id | ord_date   | ord_emp_id | ord_per_id |
|---|--------|------------|------------|------------|
| ▶ | 1      | 2012-04-10 | 2          | 1          |
| * | NULL   | NULL       | NULL       | NULL       |

### Жишээ 7:

Хамгийн бага үнэтэй, хамгийн бага тоо хэмжээтэй захиалгыг харах :

| odr_numb | odr_cus_id | odr_unit_price | odr_quantity |
|----------|------------|----------------|--------------|
| 1        | 1          | 10.00          | 1            |
| 2        | 2          | 10.00          | 3            |
| 3        | 1          | 5.00           | 1            |

```
SELECT * FROM orders
      WHERE (odr_unit_price,odr_quantity) IN
            (SELECT MIN(odr_unit_price),MIN(odr_quantity) FROM orders);
```

### Үр дүнгийн хүснэгт:

| odr_numb | odr_cus_id | odr_unit_price | odr_quantity |
|----------|------------|----------------|--------------|
| 1        | 1          | 10.00          | 1            |

### Жишээ 8 - Join:

Хамгийн их тоогоор захиалсан 3ш захиалгыг харах:

```
SELECT * FROM orders join
(SELECT buy_ord_id, buy_qty FROM buy ORDER BY buy_qty DESC LIMIT 3) as b
on orders.ord_id=b.buy_ord_id
```

### Үр дүнгийн хүснэгт:

|   | ord_id | ord_date   | ord_emp_id | ord_per_id | buy_ord_id | buy_qty |
|---|--------|------------|------------|------------|------------|---------|
| ▶ | 1      | 2012-04-10 | 2          | 1          | 1          | 2       |
|   | 1      | 2012-04-10 | 2          | 1          | 1          | 2       |
|   | 2      | 2012-04-09 | 1          | 2          | 2          | 1       |

### Жишээ 9 – Join nested:

Хамгийн их тоогоор захиалсан 3ш захиалгын ажилтанг харах:

```
SELECT * from employee join
(SELECT ord_emp_id, buy_qty FROM orders join buy on
orders.ord_id=buy.buy_ord_id ORDER BY buy_qty DESC LIMIT 3)
as b on employee.emp_id = b.ord_emp_id;
```

### Үр дүнгийн хүснэгт:

|   | emp_id | emp_name | emp_gender | emp_address | emp_phone | ord_emp_id | buy_qty |
|---|--------|----------|------------|-------------|-----------|------------|---------|
| ▶ | 2      | emp2     | 2          | mgl ub      | 8888      | 2          | 2       |
|   | 2      | emp2     | 2          | mgl ub      | 8888      | 2          | 2       |
|   | 2      | emp2     | 2          | mgl ub      | 8888      | 2          | 1       |