کارگاه

عنوان دوره: آموزش مقدماتی پایگاه‌های داده

استاد درس: دکتر توکتم خطیبی

ایمیل:toktamk57@gmail.com

فهرست فعالیت‌ها

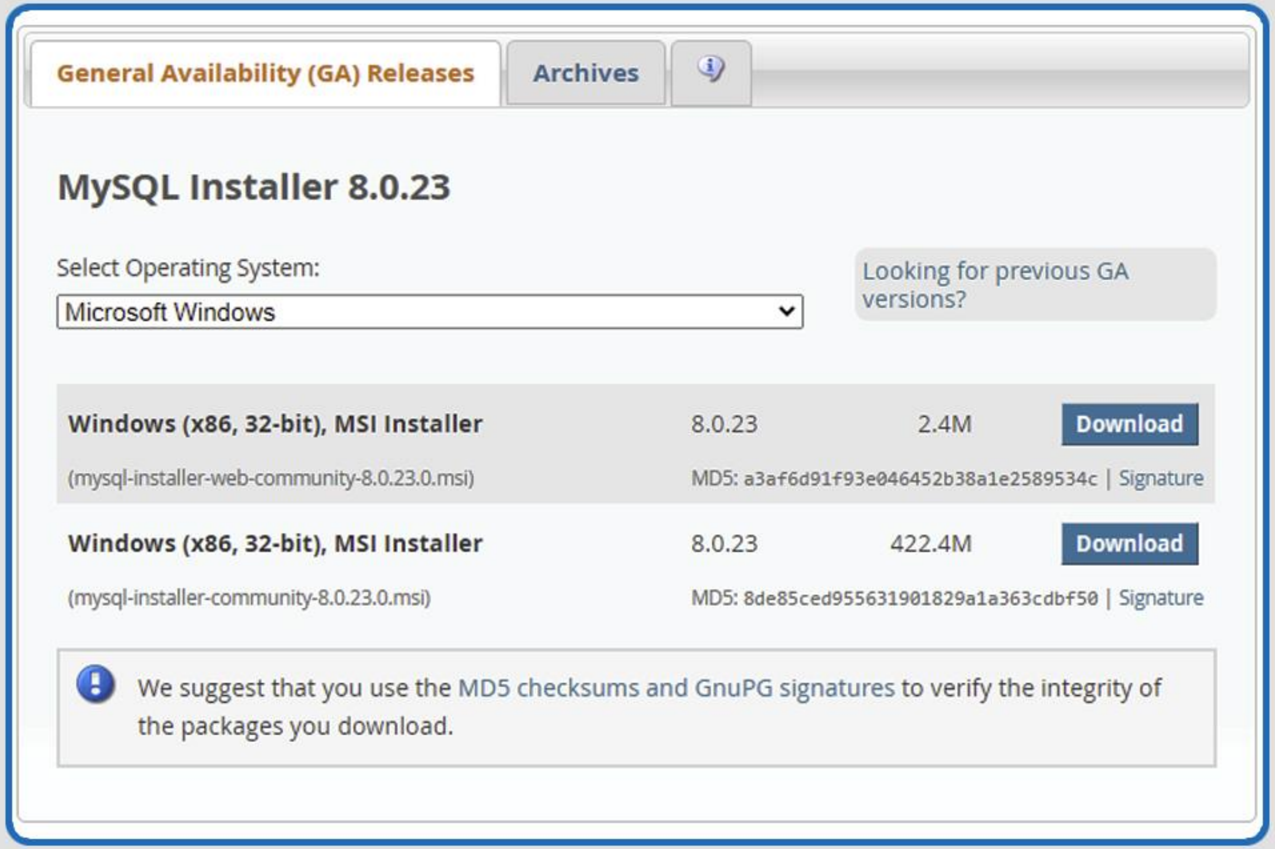
1. نحوه نصب
2. تعریف و ایجاد پایگاه داده براساس مدل منطقی
3. ایجاد جدول‌ها
4. تعریف کلیدهای اصلی
5. تعریف کلیدهای خارجی
6. درج اطلاعات در جدول‌ها
7. به روز رسانی برخی اطلاعات درج شده
8. Query های ساده بدون شرط
9. Query های دارای شرط
10. عملیات join
11. حذف برخی سطرهای جدول‌ها
12. درج اطلاعات عملیات join در یک جدول جدید

فهرست فعالیت‌ها

13. عملیات اجتماع چند Query
14. عملیات اشتراک چند Query
15. عملیات تفاضل Query
16. ایجاد View
17. تعریف توابع
18. تعریف Stored procedures
19. تعریف trigger
20. عملیات با with
21. پشتیبان گیری از پایگاه داده
22. کار با Workbench

Installation Method

The simplest and recommended method is to download MySQL Installer for Windows from https://dev.mysql.com/downloads/installer/ and execute it.


General Availability (GA) Releases
Archives
Looking for previous GA versions?
Download
Download

MySQL Installer 8.0.23

Select Operating System:
Microsoft Windows

Windows (x86, 32-bit), MSI Installer	8.0.23	2.4M	Download
(mysql-installer-web-community-8.0.23.0.msi)	MD5: a3af6d91f93e046452b38a1e2589534c | Signature		
Windows (x86, 32-bit), MSI Installer	8.0.23	422.4M	Download
(mysql-installer-community-8.0.23.0.msi)	MD5: 8de85ced955631901829a1a363cdbf50 | Signature		

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

Installation Method

- After downloading, unzip it, and double click the MSI installer .exe file.
- Then follow the steps below:

1. "Choosing a Setup Type" screen: Choose "Full" setup type. This installs all MySQL products and features. Then click the "Next" button to continue.

2. "Check Requirements" screen: The installer checks if your pc has the requirements needed. If there is some failing requirements, click on each item to try to resolve them by clicking on the Execute button that will install all requirements automatically. Click "Next".

3. "Installation" screen: See what products that will be installed. Click "Execute" to download and install the Products. After finishing the installation, click "Next".

4. "Product Configuration" screen: See what products that will be configured. Click the "MySQL Server 8.0.23" option to configure the MySQL Server. Click the "Next" button. Choose the "Standalone MySQL Server/Classic MySQL Replication" option and click on the "Next" button. In page "Type and Networking" set Config Type to "Development Computer" and "Connectivity" to "TCP/IP" and "Port" to "3006". Then, click the "Next" button.

Installation Method

5. "Authentication Method" screen: Choose "Use Strong Password Encryption for Authentication". Click "Next".

6. "Accounts and Roles" screen: Set a password for the root account. Click "Next".

7. "Windows Service" screen: Here, you configure the Windows Service to start the server. Keep the default setup, then click "Next".

8. "Apply Configuration" screen: Click the "Execute" button to apply the Server configuration. After finishing, click the "Finish" button.

9. "Product Configuration" screen: See that the Product Configuration is completed. Keep the default setting and click on the "Next" and "Finish" button to complete the MySQL package installation.

10. In the next screen, you can choose to configure the Router. Click on "Next", "Finish" and then click the "Next" button.

Installation Method

11. "Connect To Server" screen: Type in the root password (from step 6). Click the "Check" button to check if the connection is successful or not. Click on the "Next" button.

12. "Apply Configuration" screen: Select the options and click the "Execute" button.
After finishing, click the "Finish" button.

13. "Installation Complete" screen: The installation is complete. Click the "Finish" button.

Installation Method

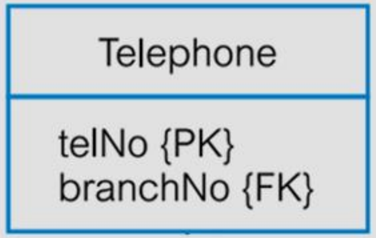
- Open the MySQL Command Line Client from cmd.
- You should see a mysql> prompt. If you have set any password, write your password here.
- Now, you are connected to the MySQL server, and you can execute all the SQL command at mysql>

Defining and Creating a Database According to the Logical Model

- create database PropertyManagementTBL;
- show databases;
- drop propertmanagement;
- use propertymanagementTbl;

Create Tables

```
create table Telephone(
        -> telNo varchar(10) Not Null,
        -> branchNo Int,
        -> primary key(telNo)
        -> );
```

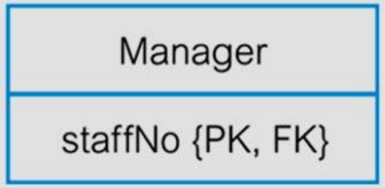


Telephone

telNo {PK}
branchNo {FK}

```
show tables;
```

```
create table Manager(
        -> staffNo Int Not null);
```

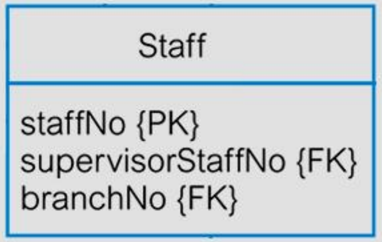


Manager

staffNo {PK, FK}

Create Tables

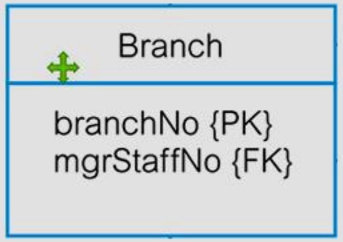
```
create table staff (
        -> staffNo Int Not Null,
        -> supervisorStaffNo Int,
        -> branchNo Int,
        -> primary key(staffNo)
        -> );
```



Staff

staffNo {PK}
supervisorStaffNo {FK}
branchNo {FK}

```
create table Branch(
        -> branchNo Int Not Null,
        -> mgrStaffNo Int,
        -> primary key(branchNo)
        -> );
```



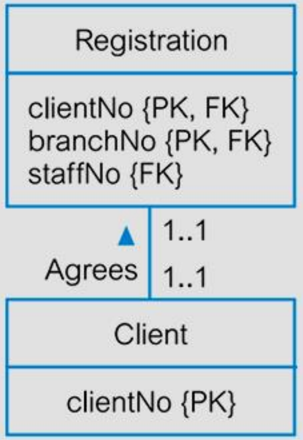
Branch

branchNo {PK}
mgrStaffNo {FK}

Create Tables

- create table Registration (

```
-> clientNo Int Not null,
-> branchNo Int Not null,
-> staffNo Int,
-> primary key(clientNo,branchNo)
-> );
```

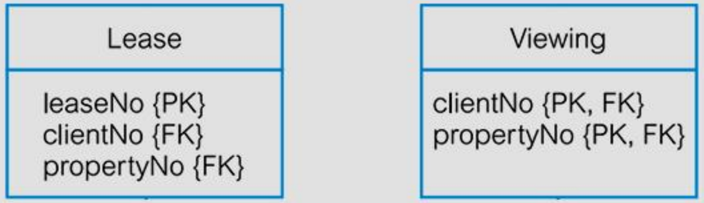


- create table client (

```
-> clientNo Int Not null);
```

Create Tables

```
create table Lease (
        -> leaseNo Int Not null,
        -> clientNo Int,
        -> propertyNo Int,
        -> primary key(leaseNo)
        -> );
```



Lease

leaseNo {PK}
clientNo {FK}
propertyNo {FK}

Viewing

clientNo {PK, FK}
propertyNo {PK, FK}

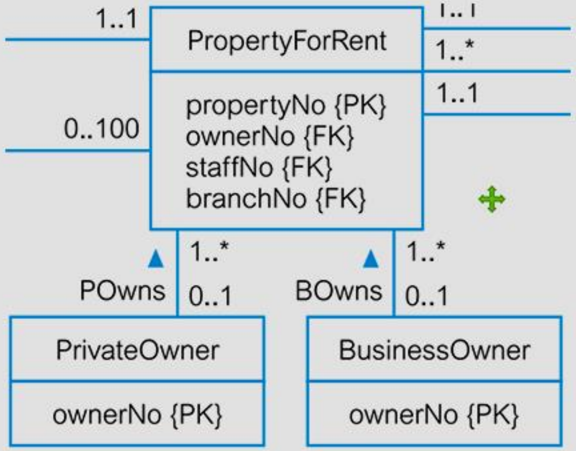
```
create table viewing (
        -> clientNo Int Not Null,
        -> PropertyNo Int Not null,
        -> primary key(clientNo, propertyNo)
        -> );
```

Create Tables

```
create table propertyForRent (

        -> propertyNo Int Not null,
        -> ownerNo Int,
        -> staffNo Int,
        -> branchNo Int,
        -> primary key(propertyNo)
        -> );
```

```
create table privateowner (

        -> ownerNo Int Not Null,
        -> primary key (ownerNo)
        -> );
```

```
create table businessOwner (

        -> ownerNo Int not null,
        -> primary key (ownerNo)
        -> );
```



Create Tables

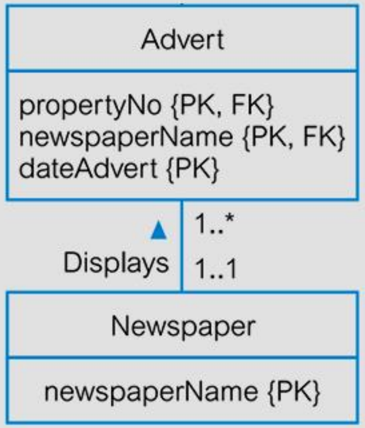
- create table Advert (

```
-> propertyNo Int Not null,
-> newspaperName Varchar(255) Not null,
-> dateAdvert DateTime,
-> primary Key(propertyNo, newspapername)
-> );
```

- create table newspaper (

```
-> newspapername Varchar(255) Not null,
-> primary key(newspapername)
-> );
```

- Show tables;



Advert

propertyNo {PK, FK}
newspaperName {PK, FK}
dateAdvert {PK}

▲ Displays

1..*

1..1

Newspaper

newspaperName {PK}

Definition of Primary Keys

- **Alter table Telephone**

  -> Add constraint FK
  -> foreign key(branchNo)
  -> References Branch(branchNo);

- **Alter table branch**

  -> Add constraint FKb
  -> foreign key(mgrstaffno)
  -> references Manager(staffno);

- **Alter table manager**

  -> Add constraint PK
  -> primary key(staffNO);

- **describe manager;**

Definition of Foreign Keys

```
alter table branch

    -> add constraint fkb
    -> foreign key (mgrstaffno)
    -> references manager(staffno);
   describe branch;
```

```
alter table staff

    -> add constraint fks
    -> foreign key (branchno)
    -> references Branch(branchno);
  describe staff;
```

```
alter table staff

    -> add constraint fkss
    -> foreign key (supervisorstaffno)
    -> references staff(staffno);
```

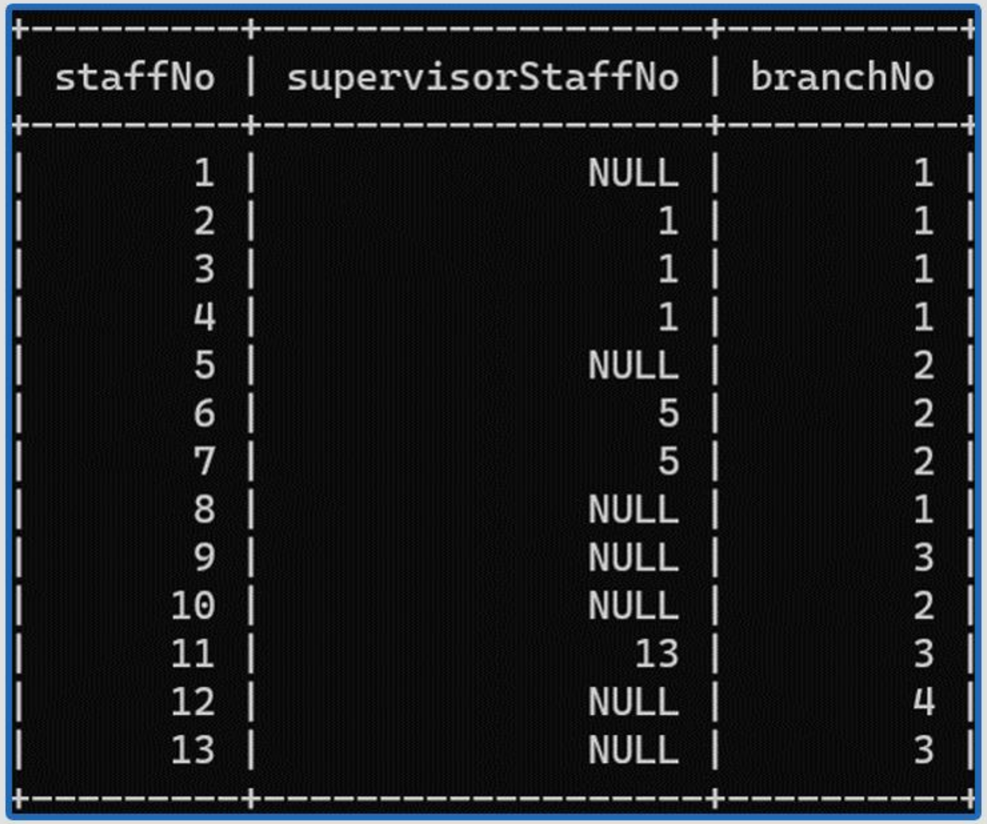
Insert Information in Tables

- insert into telephone (telno, branchno) values (111, 1);
- insert into staff (staffno, supervisorstaffno,branchno) values (1,,1);
- insert into branch (branchno, mgrstaffno) values (1,1);
- insert into branch (branchno) values (1);
- insert into branch (branchNo) values (2);
- Select * from branch;

Insert Information in Tables

Staff:

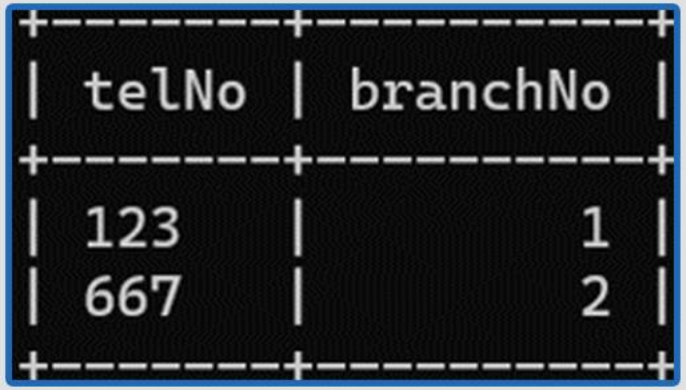
- insert into staff (staffNo, branchNo) values (1,1);



```
+---------+-------------------+----------+
| staffNo | supervisorStaffNo | branchNo |
+---------+-------------------+----------+
|       1 |              NULL |        1 |
|       2 |                 1 |        1 |
|       3 |                 1 |        1 |
|       4 |                 1 |        1 |
|       5 |              NULL |        2 |
|       6 |                 5 |        2 |
|       7 |                 5 |        2 |
|       8 |              NULL |        1 |
|       9 |              NULL |        3 |
|      10 |              NULL |        2 |
|      11 |                13 |        3 |
|      12 |              NULL |        4 |
|      13 |              NULL |        3 |
+---------+-------------------+----------+
```

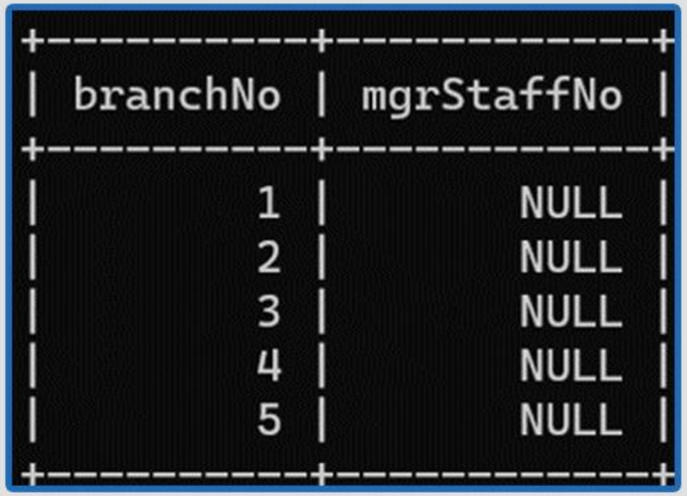
Insert Information in Tables

- **Telephone:**



```
+-------+-----------+
| telNo | branchNo  |
+-------+-----------+
| 123   |         1 |
| 667   |         2 |
+-------+-----------+
```

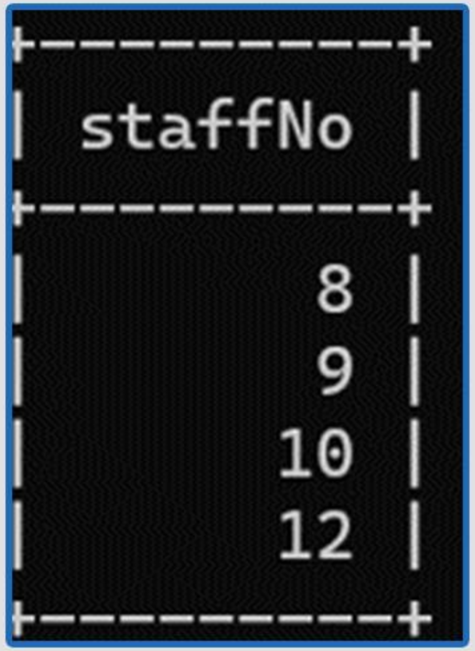
- **Branch:**



```
+----------+------------+
| branchNo | mgrStaffNo |
+----------+------------+
|        1 |       NULL |
|        2 |       NULL |
|        3 |       NULL |
|        4 |       NULL |
|        5 |       NULL |
+----------+------------+
```

Insert Information in Tables

- **Manager:**



```
+---------+
| staffNo |
+---------+
|       8 |
|       9 |
|      10 |
|      12 |
+---------+
```

Insert Information in Tables

- insert into staff (staffno, branchno) values (9,3), (11,3), (12,4);

Insert Information in Tables

```
update staff set supervisorstaffno = 13 where staffno = 11;
```

Simple Queries Without Conditions

- select * from staff;
- select * from manager;
- select * from telephone;

Conditional Queries

- select * from staff where branchno =1 ;
- select * from staff where branchno = 3;
- select * from staff where branchno = 3 and supervisorstaffno is not null;

Join Operation

- select * from staff left join manager on staff.staffno = manager.staffno;
- select * from staff left join manager on staff.staffno = manager.staffno
  where manager.staffno is not null;

Inserting Join Operation Information in a New Table

-
  ```
  insert into staffManagers select * from staff left join manager on
  staff.staffno = manager.staffno where manager.staffno is not null;
  ```

  ```
  ERROR 1146 (42S02): Table 'propertymanagement.staffmanagers' doesn't exist
  ```

-
  ```
  create table staffManager select * from staff left join manager on
  staff.staffno = manager.staffno where manager.staffno is not null;
  ```

  ```
  ERROR 1060 (42S21): Duplicate column name 'staffNo'
  ```

-
  ```
  create table staffmanager select staffno as sno, branchno
  from staff left join manager on staff.staffno = manager.
  staffno where manager.staffno is not null;
  ```

  ```
  ERROR 1052 (23000): Column 'staffno' in field list is ambiguous
  ```

Inserting Join Operation Information in a New Table

- create table staffMngr select s.staffno, s.branchno from staff s left join manager m using (staffno) where m.staffno is not null;

- Select * from staffMngr;

Deleting some Rows of Tables

- insert into telephone (telNo,branchno) values (289,3), (442,4);
- select * from telephone;
- delete from telephone where branchno = 4;

Comprehensive Multi-Query Operation

- select * from branch union select * from telephone;

- select * from staff union select * from branch;
  ERROR 1222 (21000): The used SELECT statements
  have a different number of columns

- select * from branch union select staffno,branchno from staff;

Multi-Query Subscription Operation

- select staffno from staff intersect select staffno from manager;

- select branchno from telephone intersect select distinct branchno from staff;

- select * from branch intersect select * from staff;
  ERROR 1222 (21000): The used SELECT statements
  have a different number of columns

Multi-Query Difference Operation

-
  ```
  select staffno from staff minus select staffno from manager;
  ```

-
  ```
  ERROR 1064 (42000): You have an error in your SQL syntax; check
  the manual that corresponds to your MySQL server version for the
  right syntax to use near 'select staffno from manager' at line 1
  ```

-
  ```
  select staffno from staff not exist in (select staffno from manager);
  ```

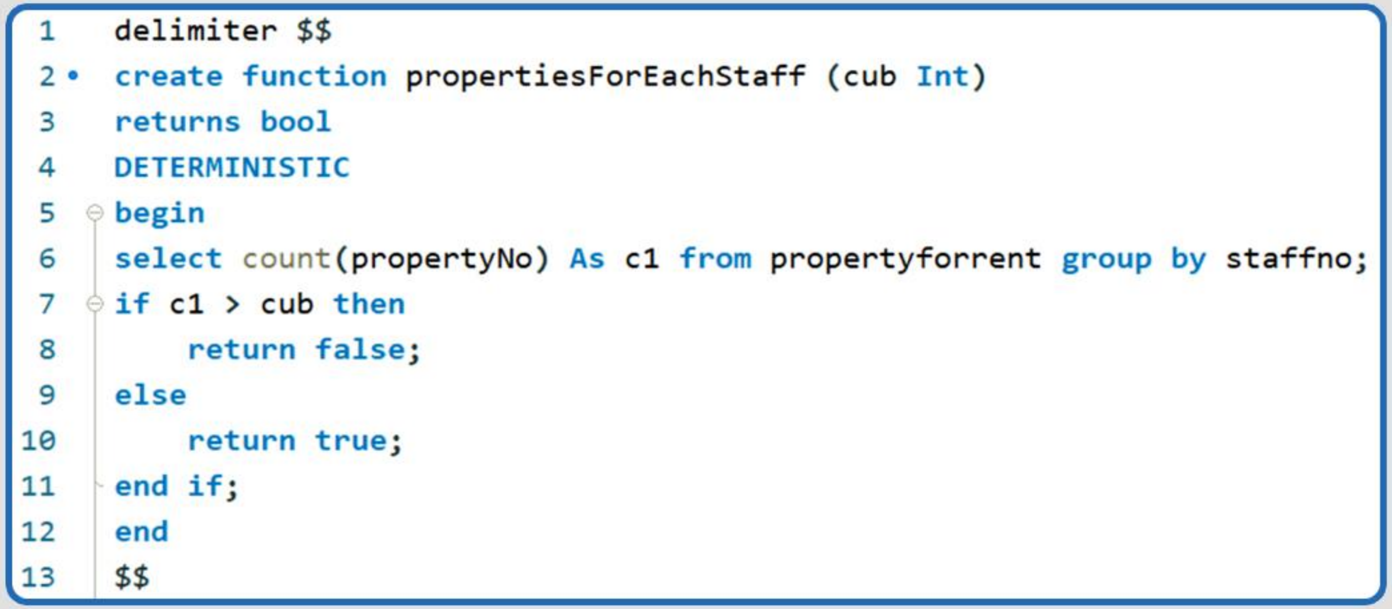
-
  ```
  ERROR 1064 (42000): You have an error in your SQL syntax;
  check the manual that corresponds to your MySQL server
  version for the right syntax to use near 'not exist in (select
  staffno from manager)' at line 1
  ```

-
  ```
  select staffno from staff except select staffno from manager;
  ```

Create View

- Create a view that lists staffno and branchno for staffs having supervisor.
- create view hasSupervisor as select staffNo, branchno from staff where supervisorstaffno is not null;

Definition of Functions



```
delimiter $$
create function propertiesForEachStaff (cub Int)
returns bool
DETERMINISTIC
begin
select count(propertyNo) As c1 from propertyforrent group by staffno;
if c1 > cub then
    return false;
else
    return true;
end if;
end
$$
```

Definition of Stored Procedures

- CREATE PROCEDURE `staffsWithMoreClients` ()

- BEGIN

- select sum(ownerNo) as so1 from propertyforrent group by staffno order by sum(ownerno) Desc;

- END

Workbench

- Connect to database
- Insert more data in all tables
- Create views
- Define stored procedure
- Define functions
- Define queries