SACRAMENTO STATE
COLLEGE OF CONTINUING EDUCATION

MSBA202

Data Management for Business Analytics

Section - 70

Prepared by – Group 1

Bindu Nithyananda – 304429021
Md Tarbiar Swat - 304454332
Hashim Gilani - 304439369
Suleman Ahmad - 304467046

Group Project - Part 1

Submission Date
April 10, 2025

Prepared for
Beom-Jin Choi

1. Business Objective

To upgrade the existing operational database for a property management company by enhancing data integrity, reducing redundancy, and providing improved support for complex queries. The upgraded system will effectively manage detailed information on buildings, apartments, managers, staff, inspections, leases, maintenance requests, and corporate clients. In addition, the solution will support an analytical database for historical analysis and decision-making, enabling the company to monitor leasing trends, maintenance costs, and overall property performance.

2. Database & Business Requirement

- **Operational Database Requirements:**
    - **Manage Buildings & Managers:**
      Record details of buildings and the managers who oversee them, including tracking which building a manager resides in.
    - **Apartment & Leasing Management:**
      Capture apartment details (number of bedrooms, occupancy status) and manage leasing information for corporate clients. This includes tracking lease start/end dates, monthly rent, and security deposits.
    - **Maintenance Requests:**
      Record maintenance requests from tenants with details such as request date, description, and status.
    - **Inspections:**
      Manage and track building inspections by inspectors, including inspection dates and next scheduled inspection dates.
    - **Cleaning Operations:**
      Track cleaning assignments by linking staff members to the apartments they clean.

3. List of Entities and Attributes

- **Building** (Building_ID,Street,State,City ZipCode,Manager_ID)
- **Manager** (Manager ID,MFull_Name,Salary, MPhone,MEmail,Building_ID)
- **Apartment** (Apartment_ID, No_of_bedrooms, Rental_status, Building_ID)
- **MaintenanceRequest** (Request_ID, Building_ID, AptNo, Request_Date, Description, Status)
- **Inspector** (Inspector_ID,InspectorName,IPhone,IEmail)
- **Inspects**(Inspector_ID,Insp_date,next_insp,Building_ID)
- **Staff** (Staff_ID, S_Name, SPhone,SEmail)
- **Cleans** (Apartment_ID, Staff_ID, Building_ID)

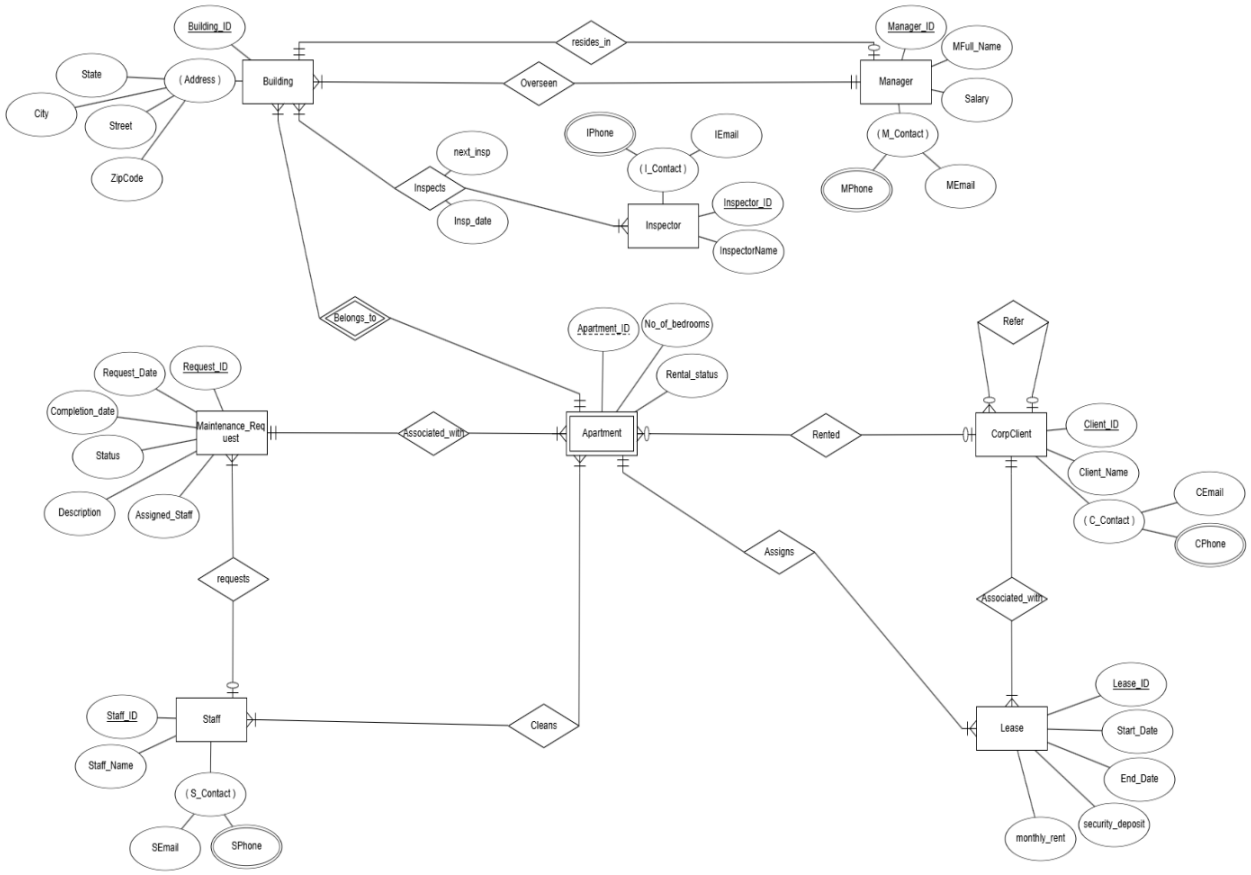
- **Resident_Feedback** (Feedback_ID, Feedback_Date, Category, Description, Status, Apartment_ID)
- **CorpClient** (Client_ID, Client_Name, CEmail,CPhone)
- **Lease** (Lease_ID, Building_ID, AptNo, Client_ID, LeaseStartDate, LeaseEndDate, MonthlyRent, SecurityDeposit)

4. ER Diagram

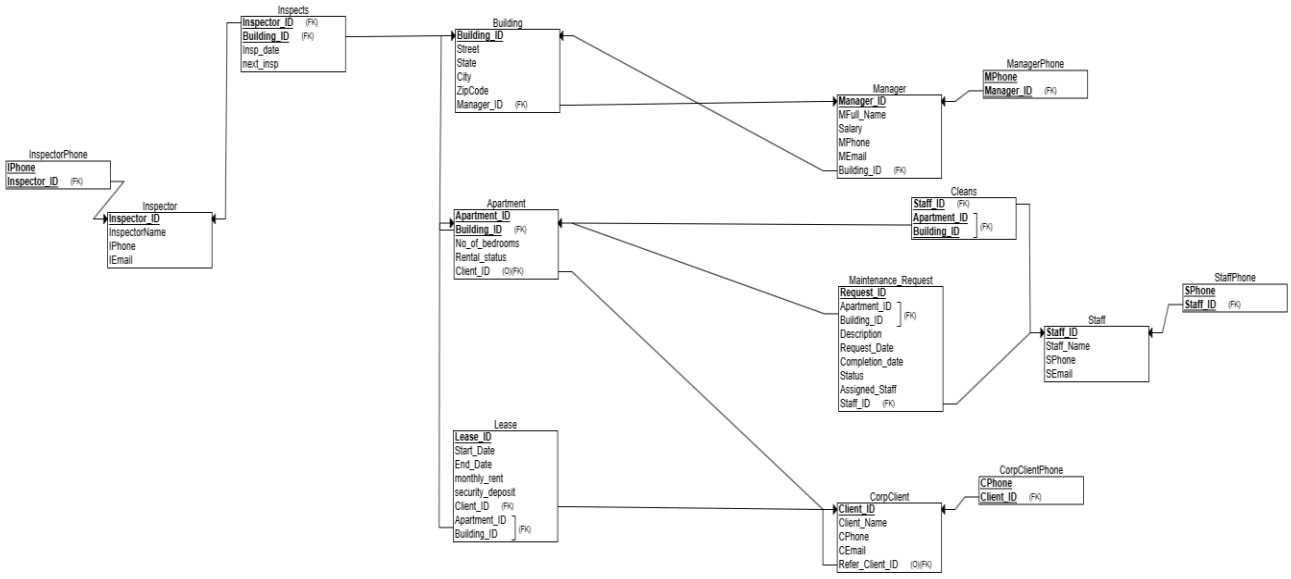
The ER Diagram visually represents the conceptual design of the property management system. It illustrates all major entities—such as Building, Manager, Apartment, MaintenanceRequest, Lease, CorpClient, Inspector, Staff, and related associative entities like Overseen, Inspection, and Cleans—and the relationships between them.

5. (20 Points) Create an ER diagram and the relational schema for the database

5.1 Entity-Relationship Diagram: PROPERTY MANAGEMENT SYSTEM



5.2 Relational-Schema: PROPERTY MANAGEMENT SYSTEM



6. (15 points) Create the data dictionary (an example is shown below) that documents metadata about tables and related constraints.

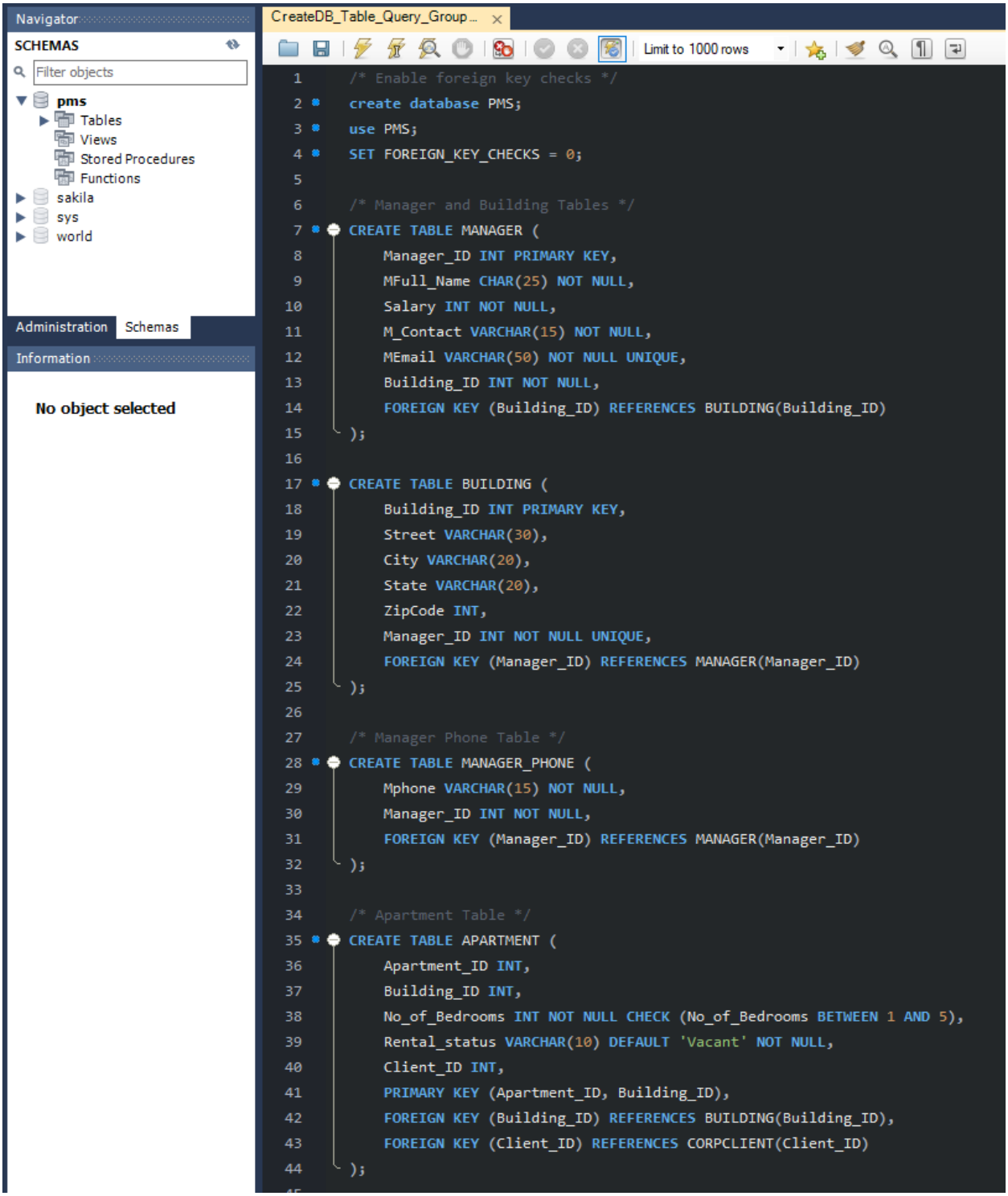
6.1 Data-Dictionary: PROPERTY MANAGEMENT SYSTEM

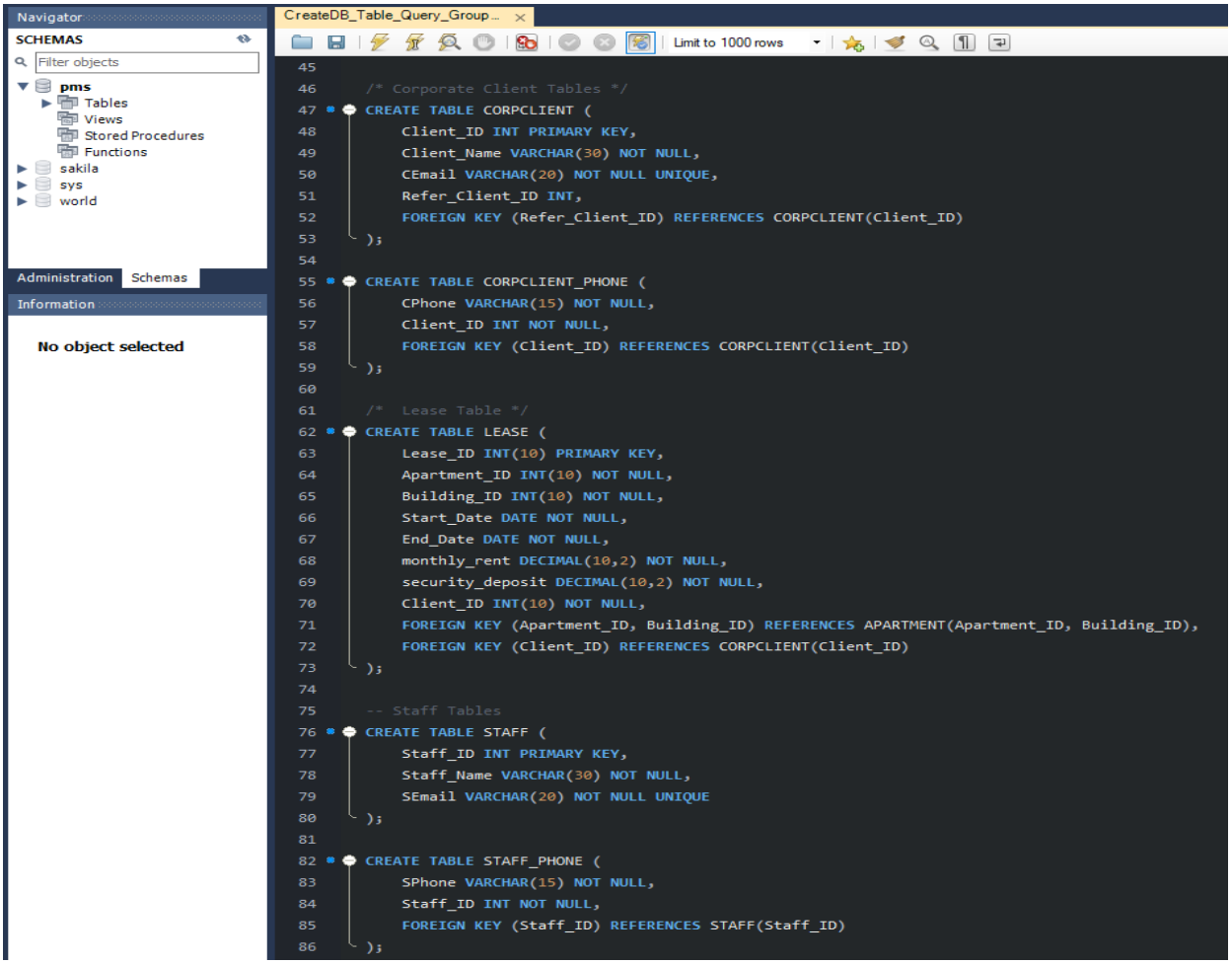
Entity:	**MANAGER TABLE**	Contains all data about property managers.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Manager_ID	Unique manager ID	INT	10 digits, unique ID	-	YES	YES	PK
MFull_Name	Manager's full name	CHAR	25 characters	-	YES	NO	-
Salary	Manager's annual salary	INT	20 digits	-	YES	NO	-
M_Contact	Manager's contact number	VARCHAR	15 alpha numeric characters	-	YES	NO	-
MEmail	Manager's email address	VARCHAR	50 alpha numeric characters		YES	YES	
Building_ID	Building where the manager resides	INT	10 digits	-	YES	NO	FK
	MANAGER PHONE TABLE						
Mphone	Phone number of Manager	INT	10 digits	-	YES	NO	Null Allowed
Manager_ID	Unique ID of Manager	INT	10 digits, unique ID	-	YES	YES	FK
Entity:	**BUILIDING TABLE**	Stores building details and assigned manager.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Building_ID	Building ID	INT	10 digits, unique ID	-	YES	YES	PK
Street	Building street address	VARCHAR	30 alpha numeric characters				
City	Building in which city	VARCHAR	20 alpha numeric characters				
State	Building in which state	VARCHAR	20 alpha numeric characters				
ZipCode	Building postal code	INT	10 digits				
Manager_ID	Assigned Manager ID	INT	10 digits, unique ID	-	YES	YES	FK
Entity:	**APARTMENT TABLE**	Tracks apartments within buildings and their status.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Apartment_ID	Apartment Number	INT	10 digits, unique ID	-	YES	YES	PK (Composite)
Building_ID	Building ID	INT	10 digits, unique ID	-	YES	NO	PK(Composite)
No_of_Bedrooms	Number of Bedrooms	INT	2 digits, (1 - 5 Bedrooms)	-	YES	NO	-
Rental_status	Occupancy Status	VARCHAR	Occupied or Vacant	Vacant	YES	NO	-
Client_ID	Corporate Client ID	INT	10 digits (if occupied)	NULL	NO	NO	FK
Entity:	**CORPCLIENT TABLE**	Tracks apartments within buildings and their status.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Client_ID	Corporate Client ID	INT	10 digits, unique ID	-	YES	YES	PK
Client_Name	Client Name	VARCHAR	30 alpha numeric characters	-	YES	NO	-
CEmail	Client Phone Number	VARCHAR	20 alpha numeric characters	-	YES	YES	-
Refer_Client_ID	Referrer Client ID	INT	10 digits (if referred)	NULL	NO	NO	FK

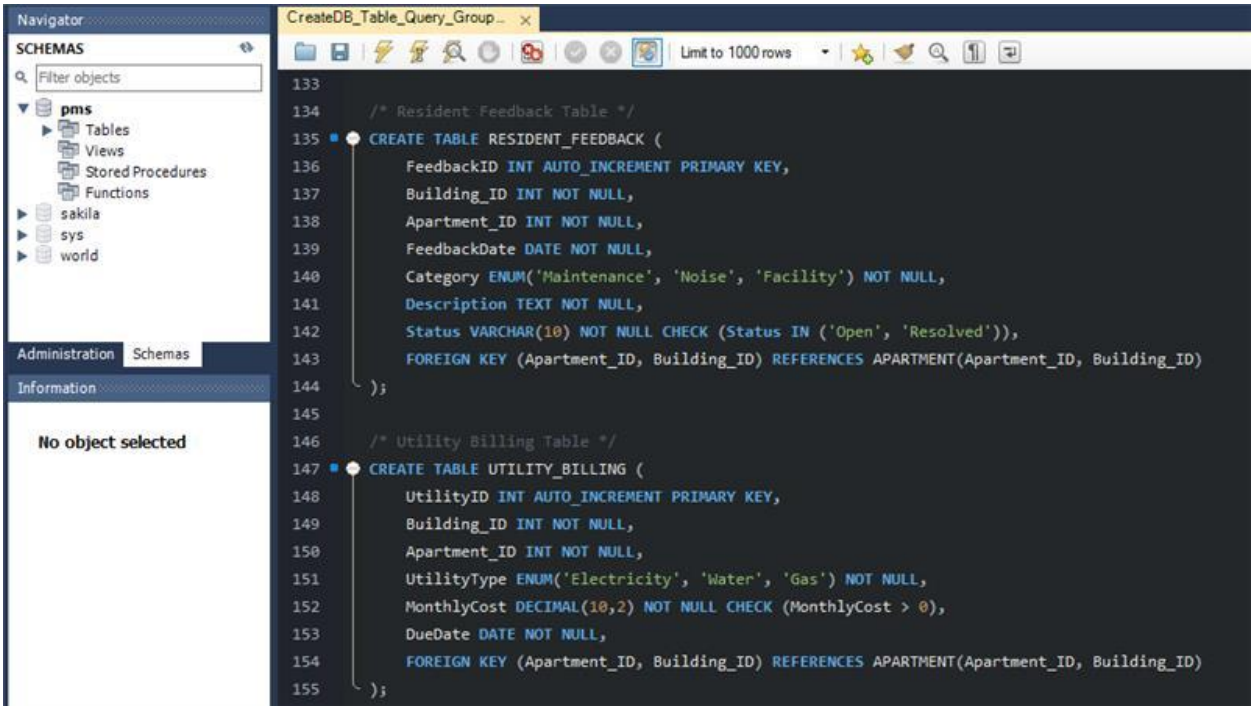
	CORPCLIENT PHONE TABLE						
CPhone	Phone number of corporate client	INT	10 digits	-	YES	NO	Null Allowed
Client_ID	Unique ID of corporate client	INT	10 digits, unique ID	-	YES	YES	FK
Entity:	**LEASE TABLE**	Tracks apartment lease agreements.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Lease_ID	Lease ID	INT	10 digits, unique ID	-	YES	YES	PK
Apartment_ID	Apartment Number	VARCHAR	10 digits	-	YES	NO	FK (Composite)
BuildingID	Building ID	INT	10 digits	-	YES	NO	FK (Composite)
Start_Date	Lease Start Date	DATE	Format: YYYY-MM-DD	-	YES	NO	-
End_Date	Lease End Date	DATE	Format: YYYY-MM-DD	-	YES	NO	-
monthly_rent	Monthly Rent	DECIMAL(10,2)	Currency	-	YES	NO	-
security_deposit	Security Deposit	DECIMAL(10,2)	Currency	-	YES	NO	-
Client_ID	Corporate Client ID	INT	10 digits	-	YES	NO	FK
Entity:	**STAFF TABLE**	Records staff members responsible for cleaning.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Staff_ID	Staff ID	INT	10 digits, unique ID	-	YES	YES	PK
Staff_Name	Staff Name	VARCHAR	30 alpha numeric characters	-	YES	NO	-
SEmail	Staff email address	VARCHAR	20 alpha numeric characters	-	YES	YES	-
	STAFF PHONE TABLE						
SPhone	Phone number of staff	INT	10 digits	-	YES	NO	Null Allowed
Staff_ID	Unique ID of staff	INT	10 digits, unique ID	-	YES	YES	FK
Entity:	**INSPECTOR TABLE**	Tracks building inspectors.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Inspector_ID	Inspector ID	INT	10 digits, unique ID	-	YES	YES	PK
InspectorName	Inspector Name	VARCHAR	30 alpha numeric characters	-	YES	NO	-
IEmail	Inspector email address	VARCHAR	20 alpha numeric characters	-	YES	YES	-
	INSPECTOR PHONE TABLE						
IPhone	Phone number of inspector	INT	10 digits	-	YES	NO	Null Allowed
Inspector_ID	Unique ID of inspector	INT	10 digits, unique ID	-	YES	YES	FK
Entity:	**INSPECTS TABLE**	Logs building inspections and schedules.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Inspector_ID	Inspector ID	INT	10 digits, unique ID	-	YES	YES	PK
Building_ID	Building ID	INT	10 digits	-	YES	NO	FK
Insp_date	Inspection date	DATE	Format: YYYY-MM-DD	-	YES	NO	-
Insp_date	Inspection date	DATE	Format: YYYY-MM-DD	-	YES	NO	-
next_insp	Next inspection date	DATE	Format: YYYY-MM-DD	-	YES	NO	-
Entity:	**CLEANS TABLE**	Assigns staff to clean specific apartments.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Staff_ID	Identifier of the staff member	INT	10 digits	-	YES	NO	FK
Apartment_ID	Identifier of the apartment	INT	10 digits	-	YES	NO	FK (Composite)
Building_ID	Building ID	INT	10 digits	-	YES	NO	FK (Composite)
Entity:	**MAINTENANCE REQUEST TABLE**	Records tenant maINTenance requests.					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
Request_ID	Unique id for each request	INT	10 digits, unique ID	-	YES	YES	PK
Building_ID	Identifier of the building	INT	10 digits	-	YES	NO	FK (Composite)
Apartment_ID	Apartment Number	INT	10 digits	-	YES	NO	FK (Composite)
Request_Date	Date of Request	DATE	Format: YYYY-MM-DD	-	YES	NO	-
Completion_date	Date of completion	DATE	Format: YYYY-MM-DD	-	NO	NO	
Status	Maintenance status	VARCHAR	20 alpha numeric characters	-	YES	NO	
Assigned_Staff	Staff Member Assigned	VARCHAR	30 alpha numeric characters	-	YES	NO	FK
Staff_ID	Request Status	INT	10 digits	-	NO	NO	-
Entity:	**RESIDENT FEEDBACK TABLE**	Track utility billing of apartments					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
FeedbackID	Unique feedback ID	INT	Auto-increment	-	YES	YES	PK
BuildingID	Building ID	INT	Links to Apartment	-	YES	NO	-
AptNo	Apartment number	INT	Links to Apartment	-	YES	NO	-
FeedbackDate	Date of feedback submission	DATE	Format: YYYY-MM-DD	-	YES	NO	-
Category	Feedback category	ENUM	MaINTenance, Noise, Facility	-	YES	NO	-
Description	Feedback details	TEXT		-	YES	NO	-
Status	Feedback resolution status	VARCHAR(10)	CHECK: Open/Resolved	-	YES	NO	-
Entity:	**UTILITY BILLING**	Track feedbacks from residents					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
UtilityID	Unique utility bill ID	INT	Auto-increment	-	YES	YES	PK
BuildingID	Building ID	INT	Links to Apartment	-	YES	NO	-
AptNo	Apartment number	INT	Links to Apartment	-	YES	NO	-
UtilityType	Type of utility	ENUM	Electricity, Water, Gas	-	YES	NO	-
MonthlyCost	Monthly utility cost	DECIMAL(10,2)	Positive value (e.g., 150.00)	-	YES	NO	-
DueDate	Pavment due date	DATE	Format: YYYY-MM-DD	-	YES	NO	-

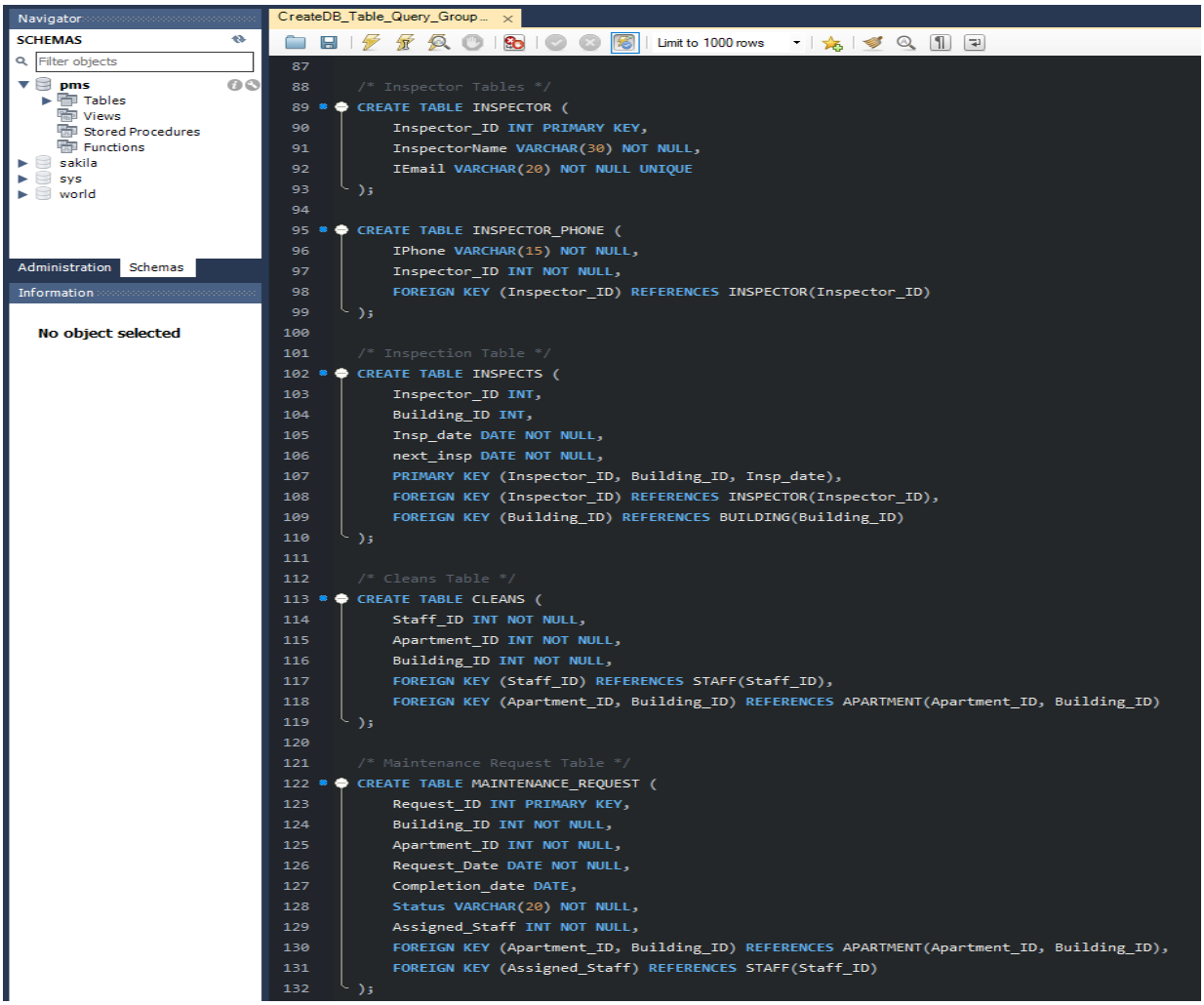
7. (10 points) Populate the database with sample data using SQL DDL codes (at least 10 rows).

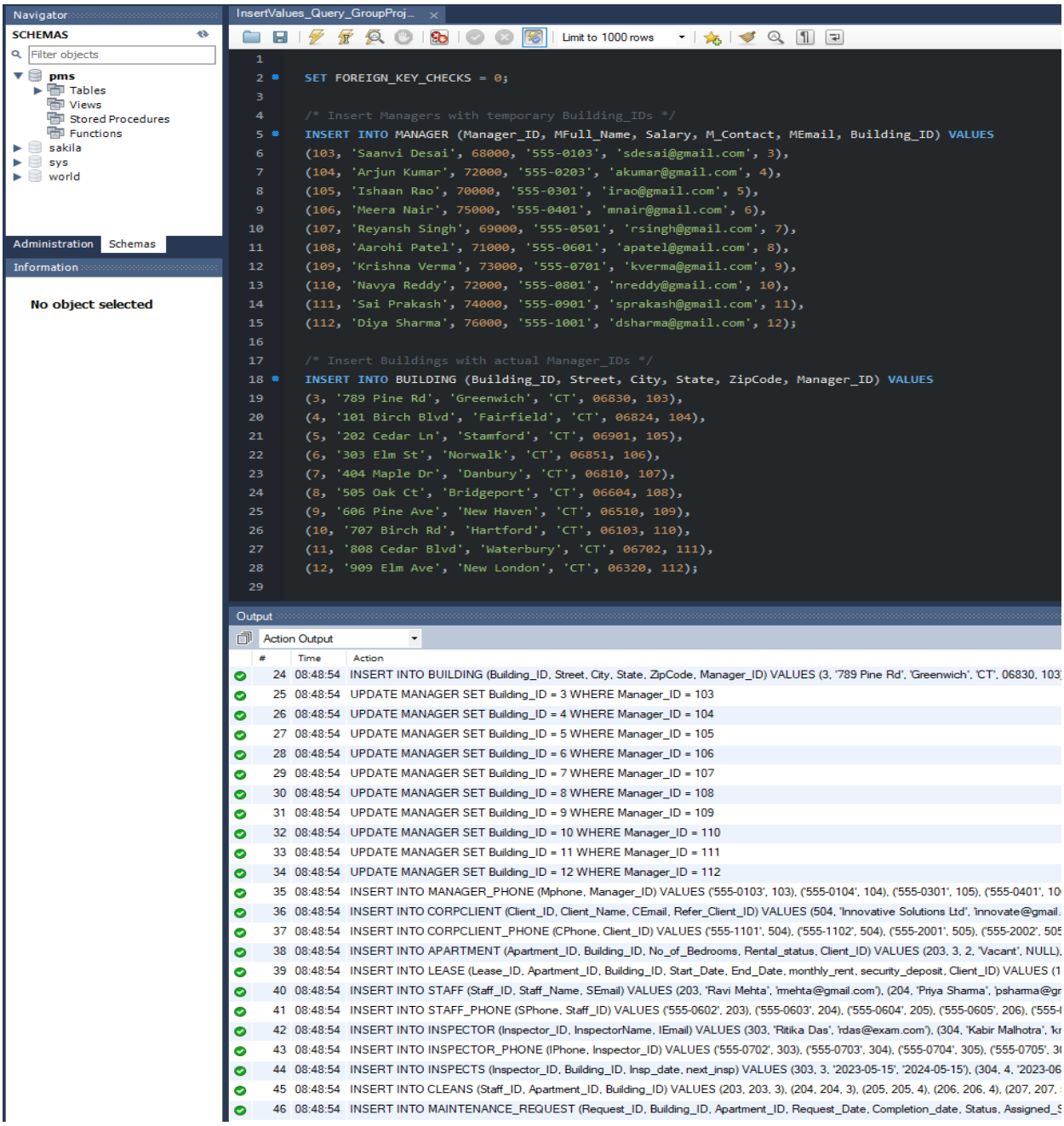
7.1 SQL Database creation and table population



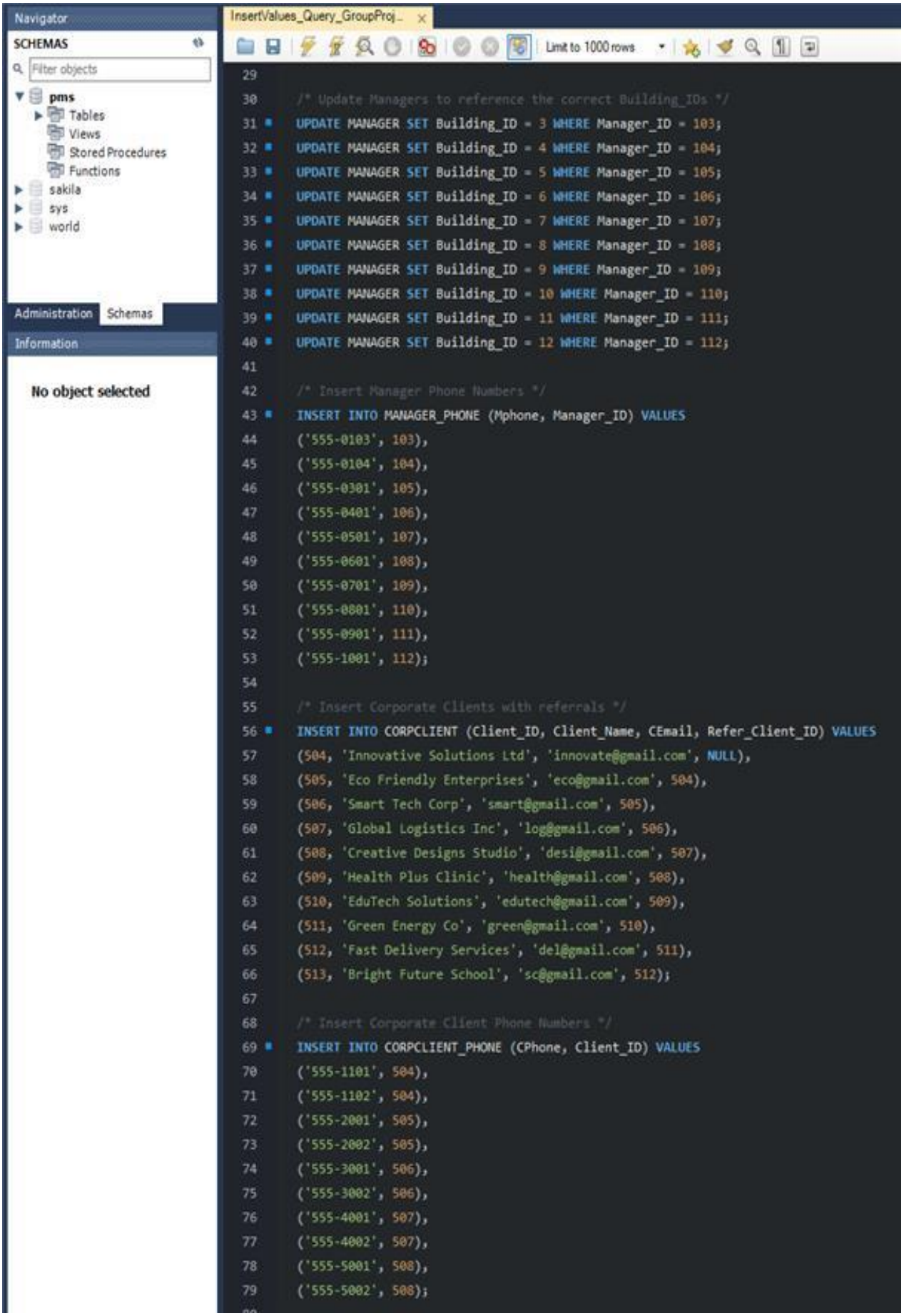

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
CreateDB_Table_Query_Group...
Limit to 1000 rows
/* Corporate Client Tables */
CREATE TABLE CORPCLIENT (
Client_ID INT PRIMARY KEY,
Client_Name VARCHAR(30) NOT NULL,
CEmail VARCHAR(20) NOT NULL UNIQUE,
Refer_Client_ID INT,
FOREIGN KEY (Refer_Client_ID) REFERENCES CORPCLIENT(Client_ID)
);
CREATE TABLE CORPCLIENT_PHONE (
CPhone VARCHAR(15) NOT NULL,
Client_ID INT NOT NULL,
FOREIGN KEY (Client_ID) REFERENCES CORPCLIENT(Client_ID)
);
/* Lease Table */
CREATE TABLE LEASE (
Lease_ID INT(10) PRIMARY KEY,
Apartment_ID INT(10) NOT NULL,
Building_ID INT(10) NOT NULL,
Start_Date DATE NOT NULL,
End_Date DATE NOT NULL,
monthly_rent DECIMAL(10,2) NOT NULL,
security_deposit DECIMAL(10,2) NOT NULL,
Client_ID INT(10) NOT NULL,
FOREIGN KEY (Apartment_ID, Building_ID) REFERENCES APARTMENT(Apartment_ID, Building_ID),
FOREIGN KEY (Client_ID) REFERENCES CORPCLIENT(Client_ID)
);
-- Staff Tables
CREATE TABLE STAFF (
Staff_ID INT PRIMARY KEY,
Staff_Name VARCHAR(30) NOT NULL,
SEmail VARCHAR(20) NOT NULL UNIQUE
);
CREATE TABLE STAFF_PHONE (
SPhone VARCHAR(15) NOT NULL,
Staff_ID INT NOT NULL,
FOREIGN KEY (Staff_ID) REFERENCES STAFF(Staff_ID)
);

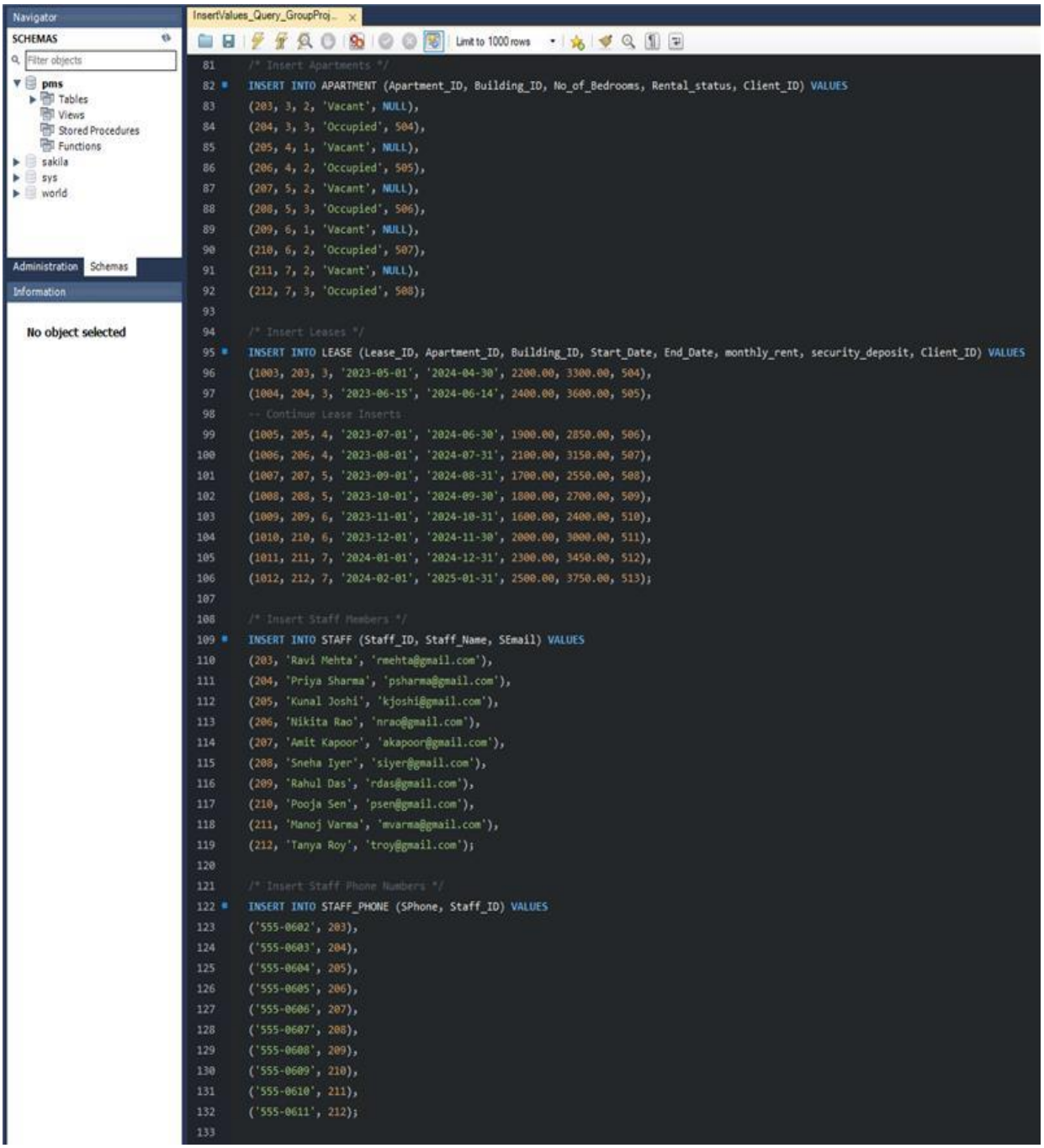

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
CreateDB_Table_Query_Group...
Limit to 1000 rows
/* Resident Feedback Table */
CREATE TABLE RESIDENT_FEEDBACK (
FeedbackID INT AUTO_INCREMENT PRIMARY KEY,
Building_ID INT NOT NULL,
Apartment_ID INT NOT NULL,
FeedbackDate DATE NOT NULL,
Category ENUM('Maintenance', 'Noise', 'Facility') NOT NULL,
Description TEXT NOT NULL,
Status VARCHAR(10) NOT NULL CHECK (Status IN ('Open', 'Resolved')),
FOREIGN KEY (Apartment_ID, Building_ID) REFERENCES APARTMENT(Apartment_ID, Building_ID)
);
/* Utility Billing Table */
CREATE TABLE UTILITY_BILLING (
UtilityID INT AUTO_INCREMENT PRIMARY KEY,
Building_ID INT NOT NULL,
Apartment_ID INT NOT NULL,
UtilityType ENUM('Electricity', 'Water', 'Gas') NOT NULL,
MonthlyCost DECIMAL(10,2) NOT NULL CHECK (MonthlyCost > 0),
DueDate DATE NOT NULL,
FOREIGN KEY (Apartment_ID, Building_ID) REFERENCES APARTMENT(Apartment_ID, Building_ID)
);

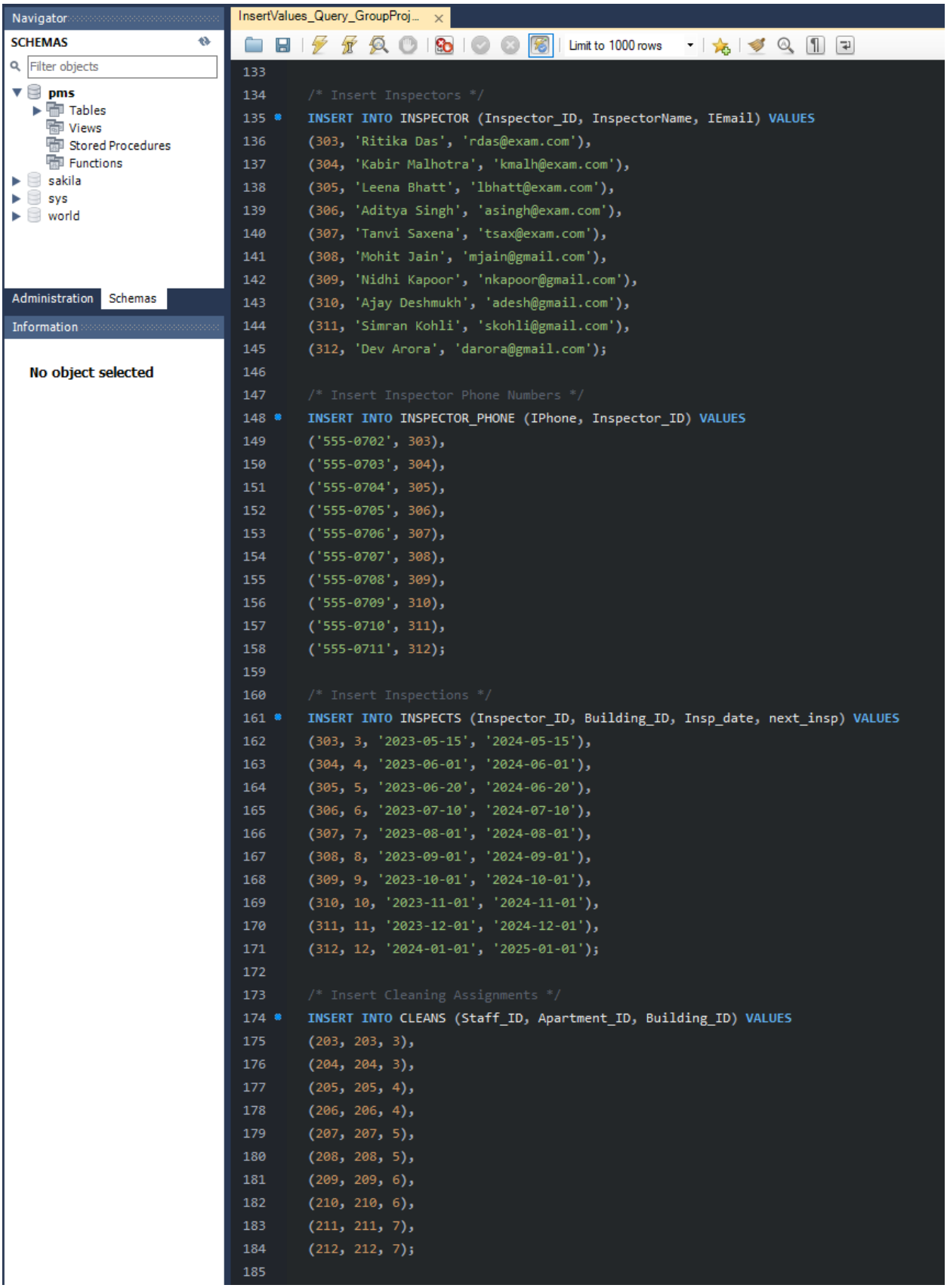

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
CreateDB_Table_Query_Group...
Limit to 1000 rows
87
88 /* Inspector Tables */
89 CREATE TABLE INSPECTOR (
90 Inspector_ID INT PRIMARY KEY,
91 InspectorName VARCHAR(30) NOT NULL,
92 IEmail VARCHAR(20) NOT NULL UNIQUE
93);
94
95 CREATE TABLE INSPECTOR_PHONE (
96 IPhone VARCHAR(15) NOT NULL,
97 Inspector_ID INT NOT NULL,
98 FOREIGN KEY (Inspector_ID) REFERENCES INSPECTOR(Inspector_ID)
99);
100
101 /* Inspection Table */
102 CREATE TABLE INSPECTS (
103 Inspector_ID INT,
104 Building_ID INT,
105 Insp_date DATE NOT NULL,
106 next_insp DATE NOT NULL,
107 PRIMARY KEY (Inspector_ID, Building_ID, Insp_date),
108 FOREIGN KEY (Inspector_ID) REFERENCES INSPECTOR(Inspector_ID),
109 FOREIGN KEY (Building_ID) REFERENCES BUILDING(Building_ID)
110);
111
112 /* Cleans Table */
113 CREATE TABLE CLEANS (
114 Staff_ID INT NOT NULL,
115 Apartment_ID INT NOT NULL,
116 Building_ID INT NOT NULL,
117 FOREIGN KEY (Staff_ID) REFERENCES STAFF(Staff_ID),
118 FOREIGN KEY (Apartment_ID, Building_ID) REFERENCES APARTMENT(Apartment_ID, Building_ID)
119);
120
121 /* Maintenance Request Table */
122 CREATE TABLE MAINTENANCE_REQUEST (
123 Request_ID INT PRIMARY KEY,
124 Building_ID INT NOT NULL,
125 Apartment_ID INT NOT NULL,
126 Request_Date DATE NOT NULL,
127 Completion_date DATE,
128 Status VARCHAR(20) NOT NULL,
129 Assigned_Staff INT NOT NULL,
130 FOREIGN KEY (Apartment_ID, Building_ID) REFERENCES APARTMENT(Apartment_ID, Building_ID),
131 FOREIGN KEY (Assigned_Staff) REFERENCES STAFF(Staff_ID)
132);

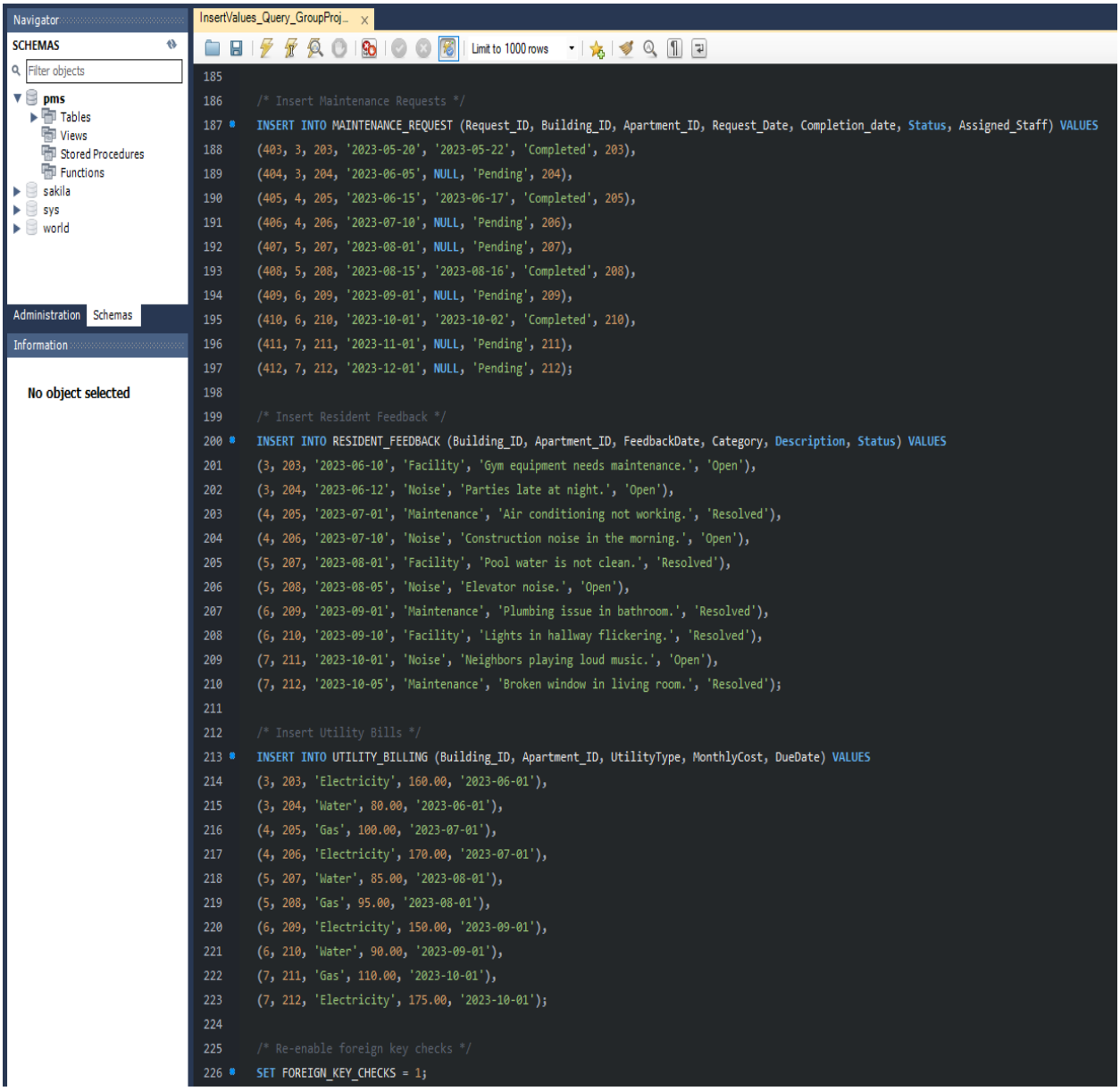


ACTION OUTPUT: SUCCESSFUL QUERY EXECUTION


Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
InsertValues_Query_GroupProj...
Limit to 1000 rows
/* Update Managers to reference the correct Building_IDs */
UPDATE MANAGER SET Building_ID = 3 WHERE Manager_ID = 103;
UPDATE MANAGER SET Building_ID = 4 WHERE Manager_ID = 104;
UPDATE MANAGER SET Building_ID = 5 WHERE Manager_ID = 105;
UPDATE MANAGER SET Building_ID = 6 WHERE Manager_ID = 106;
UPDATE MANAGER SET Building_ID = 7 WHERE Manager_ID = 107;
UPDATE MANAGER SET Building_ID = 8 WHERE Manager_ID = 108;
UPDATE MANAGER SET Building_ID = 9 WHERE Manager_ID = 109;
UPDATE MANAGER SET Building_ID = 10 WHERE Manager_ID = 110;
UPDATE MANAGER SET Building_ID = 11 WHERE Manager_ID = 111;
UPDATE MANAGER SET Building_ID = 12 WHERE Manager_ID = 112;
/* Insert Manager Phone Numbers */
INSERT INTO MANAGER_PHONE (Mphone, Manager_ID) VALUES
('555-0103', 103),
('555-0104', 104),
('555-0301', 105),
('555-0401', 106),
('555-0501', 107),
('555-0601', 108),
('555-0701', 109),
('555-0801', 110),
('555-0901', 111),
('555-1001', 112);
/* Insert Corporate Clients with referrals */
INSERT INTO CORPCLIENT (Client_ID, Client_Name, CEmail, Refer_Client_ID) VALUES
(504, 'Innovative Solutions Ltd', 'innovate@gmail.com', NULL),
(505, 'Eco Friendly Enterprises', 'eco@gmail.com', 504),
(506, 'Smart Tech Corp', 'smart@gmail.com', 505),
(507, 'Global Logistics Inc', 'log@gmail.com', 506),
(508, 'Creative Designs Studio', 'desi@gmail.com', 507),
(509, 'Health Plus Clinic', 'health@gmail.com', 508),
(510, 'EduTech Solutions', 'edutech@gmail.com', 509),
(511, 'Green Energy Co', 'green@gmail.com', 510),
(512, 'Fast Delivery Services', 'del@gmail.com', 511),
(513, 'Bright Future School', 'sc@gmail.com', 512);
/* Insert Corporate Client Phone Numbers */
INSERT INTO CORPCLIENT_PHONE (CPhone, Client_ID) VALUES
('555-1101', 504),
('555-1102', 504),
('555-2001', 505),
('555-2002', 505),
('555-3001', 506),
('555-3002', 506),
('555-4001', 507),
('555-4002', 507),
('555-5001', 508),
('555-5002', 508);

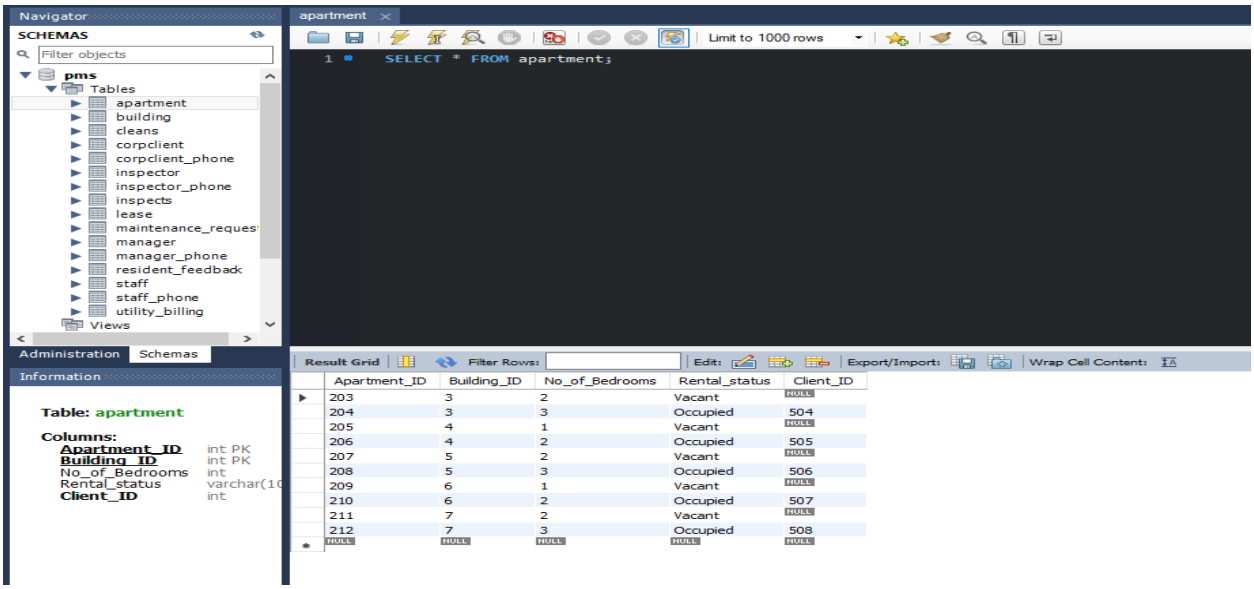

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
InsertValues_Query_GroupProj...
Limit to 1000 rows
/* Insert Apartments */
INSERT INTO APARTMENT (Apartment_ID, Building_ID, No_of_Bedrooms, Rental_status, Client_ID) VALUES
(203, 3, 2, 'Vacant', NULL),
(204, 3, 3, 'Occupied', 504),
(205, 4, 1, 'Vacant', NULL),
(206, 4, 2, 'Occupied', 505),
(207, 5, 2, 'Vacant', NULL),
(208, 5, 3, 'Occupied', 506),
(209, 6, 1, 'Vacant', NULL),
(210, 6, 2, 'Occupied', 507),
(211, 7, 2, 'Vacant', NULL),
(212, 7, 3, 'Occupied', 508);
/* Insert Leases */
INSERT INTO LEASE (Lease_ID, Apartment_ID, Building_ID, Start_Date, End_Date, monthly_rent, security_deposit, Client_ID) VALUES
(1003, 203, 3, '2023-05-01', '2024-04-30', 2200.00, 3300.00, 504),
(1004, 204, 3, '2023-06-15', '2024-06-14', 2400.00, 3600.00, 505),
-- Continue Lease Inserts
(1005, 205, 4, '2023-07-01', '2024-06-30', 1900.00, 2850.00, 506),
(1006, 206, 4, '2023-08-01', '2024-07-31', 2100.00, 3150.00, 507),
(1007, 207, 5, '2023-09-01', '2024-08-31', 1700.00, 2550.00, 508),
(1008, 208, 5, '2023-10-01', '2024-09-30', 1800.00, 2700.00, 509),
(1009, 209, 6, '2023-11-01', '2024-10-31', 1600.00, 2400.00, 510),
(1010, 210, 6, '2023-12-01', '2024-11-30', 2000.00, 3000.00, 511),
(1011, 211, 7, '2024-01-01', '2024-12-31', 2300.00, 3450.00, 512),
(1012, 212, 7, '2024-02-01', '2025-01-31', 2500.00, 3750.00, 513);
/* Insert Staff Members */
INSERT INTO STAFF (Staff_ID, Staff_Name, SEmail) VALUES
(203, 'Ravi Mehta', 'rmehta@gmail.com'),
(204, 'Priya Sharma', 'psharma@gmail.com'),
(205, 'Kunal Joshi', 'kjoshi@gmail.com'),
(206, 'Nikita Rao', 'nrao@gmail.com'),
(207, 'Amit Kapoor', 'akapoor@gmail.com'),
(208, 'Sneha Iyer', 'siyer@gmail.com'),
(209, 'Rahul Das', 'rdas@gmail.com'),
(210, 'Pooja Sen', 'psen@gmail.com'),
(211, 'Manoj Varma', 'mvarma@gmail.com'),
(212, 'Tanya Roy', 'troy@gmail.com');
/* Insert Staff Phone Numbers */
INSERT INTO STAFF_PHONE (SPhone, Staff_ID) VALUES
('555-0602', 203),
('555-0603', 204),
('555-0604', 205),
('555-0605', 206),
('555-0606', 207),
('555-0607', 208),
('555-0608', 209),
('555-0609', 210),
('555-0610', 211),
('555-0611', 212);

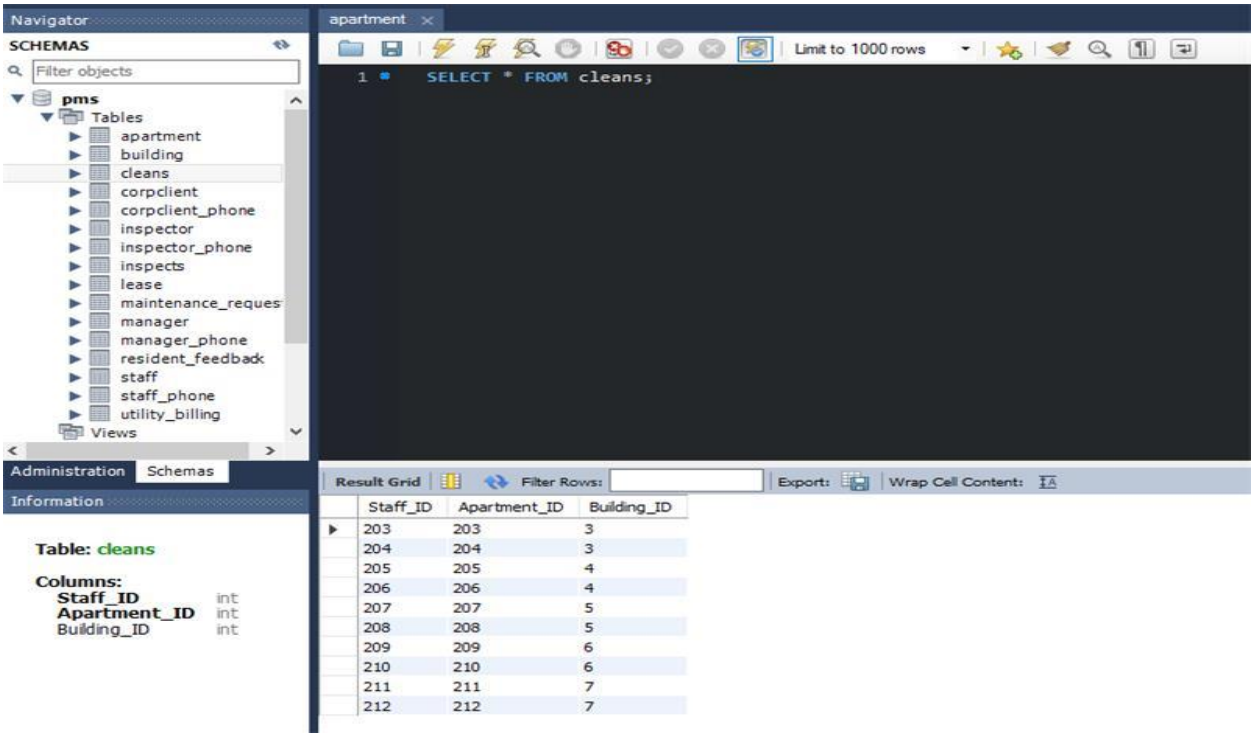

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
InsertValues_Query_GroupProj...
Limit to 1000 rows
133
134 /* Insert Inspectors */
135 INSERT INTO INSPECTOR (Inspector_ID, InspectorName, IEmail) VALUES
136 (303, 'Ritika Das', 'rdas@exam.com'),
137 (304, 'Kabir Malhotra', 'kmalh@exam.com'),
138 (305, 'Leena Bhatt', 'lbhatt@exam.com'),
139 (306, 'Aditya Singh', 'asingh@exam.com'),
140 (307, 'Tanvi Saxena', 'tsax@exam.com'),
141 (308, 'Mohit Jain', 'mjain@gmail.com'),
142 (309, 'Nidhi Kapoor', 'nkapoor@gmail.com'),
143 (310, 'Ajay Deshmukh', 'adesh@gmail.com'),
144 (311, 'Simran Kohli', 'skohli@gmail.com'),
145 (312, 'Dev Arora', 'darora@gmail.com');
146
147 /* Insert Inspector Phone Numbers */
148 INSERT INTO INSPECTOR_PHONE (IPhone, Inspector_ID) VALUES
149 ('555-0702', 303),
150 ('555-0703', 304),
151 ('555-0704', 305),
152 ('555-0705', 306),
153 ('555-0706', 307),
154 ('555-0707', 308),
155 ('555-0708', 309),
156 ('555-0709', 310),
157 ('555-0710', 311),
158 ('555-0711', 312);
159
160 /* Insert Inspections */
161 INSERT INTO INSPECTS (Inspector_ID, Building_ID, Insp_date, next_insp) VALUES
162 (303, 3, '2023-05-15', '2024-05-15'),
163 (304, 4, '2023-06-01', '2024-06-01'),
164 (305, 5, '2023-06-20', '2024-06-20'),
165 (306, 6, '2023-07-10', '2024-07-10'),
166 (307, 7, '2023-08-01', '2024-08-01'),
167 (308, 8, '2023-09-01', '2024-09-01'),
168 (309, 9, '2023-10-01', '2024-10-01'),
169 (310, 10, '2023-11-01', '2024-11-01'),
170 (311, 11, '2023-12-01', '2024-12-01'),
171 (312, 12, '2024-01-01', '2025-01-01');
172
173 /* Insert Cleaning Assignments */
174 INSERT INTO CLEANS (Staff_ID, Apartment_ID, Building_ID) VALUES
175 (203, 203, 3),
176 (204, 204, 3),
177 (205, 205, 4),
178 (206, 206, 4),
179 (207, 207, 5),
180 (208, 208, 5),
181 (209, 209, 6),
182 (210, 210, 6),
183 (211, 211, 7),
184 (212, 212, 7);
185

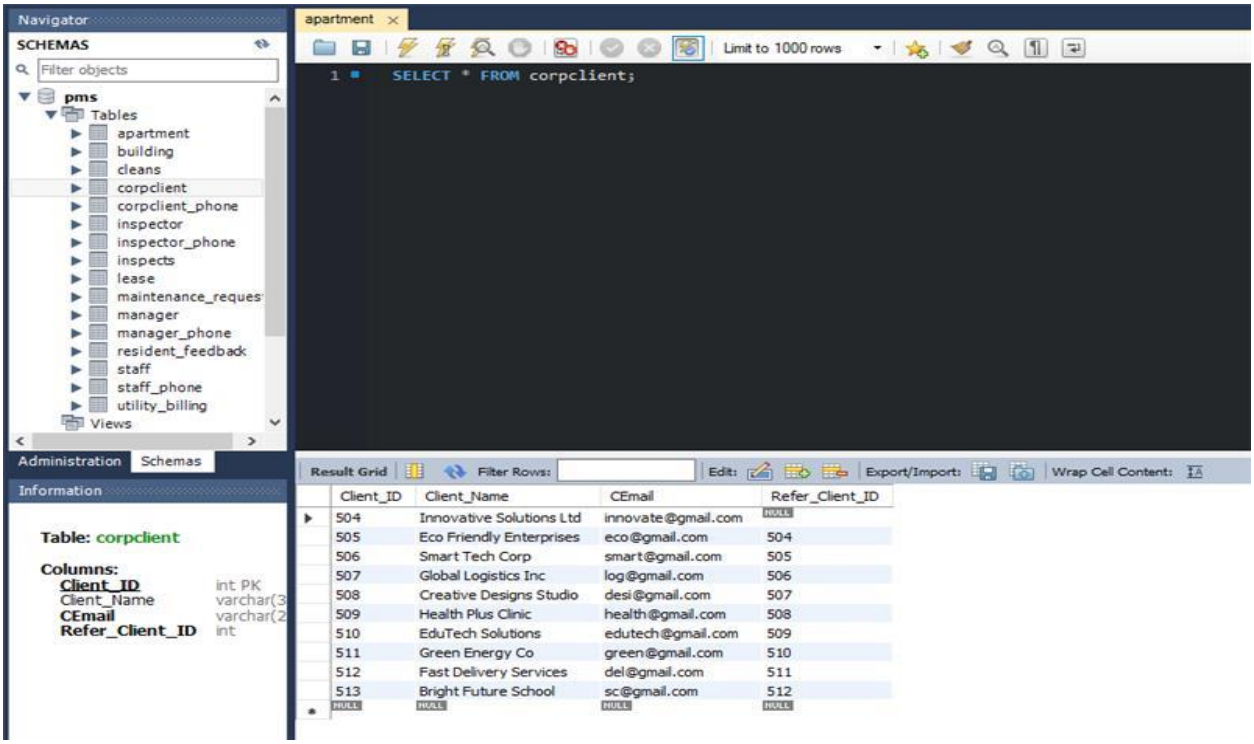

Navigator
SCHEMAS
Filter objects
pms
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Administration
Schemas
Information
No object selected
InsertValues_Query_GroupProj...
Limit to 1000 rows
/* Insert Maintenance Requests */
INSERT INTO MAINTENANCE_REQUEST (Request_ID, Building_ID, Apartment_ID, Request_Date, Completion_date, Status, Assigned_Staff) VALUES
(403, 3, 203, '2023-05-20', '2023-05-22', 'Completed', 203),
(404, 3, 204, '2023-06-05', NULL, 'Pending', 204),
(405, 4, 205, '2023-06-15', '2023-06-17', 'Completed', 205),
(406, 4, 206, '2023-07-10', NULL, 'Pending', 206),
(407, 5, 207, '2023-08-01', NULL, 'Pending', 207),
(408, 5, 208, '2023-08-15', '2023-08-16', 'Completed', 208),
(409, 6, 209, '2023-09-01', NULL, 'Pending', 209),
(410, 6, 210, '2023-10-01', '2023-10-02', 'Completed', 210),
(411, 7, 211, '2023-11-01', NULL, 'Pending', 211),
(412, 7, 212, '2023-12-01', NULL, 'Pending', 212);
/* Insert Resident Feedback */
INSERT INTO RESIDENT_FEEDBACK (Building_ID, Apartment_ID, FeedbackDate, Category, Description, Status) VALUES
(3, 203, '2023-06-10', 'Facility', 'Gym equipment needs maintenance.', 'Open'),
(3, 204, '2023-06-12', 'Noise', 'Parties late at night.', 'Open'),
(4, 205, '2023-07-01', 'Maintenance', 'Air conditioning not working.', 'Resolved'),
(4, 206, '2023-07-10', 'Noise', 'Construction noise in the morning.', 'Open'),
(5, 207, '2023-08-01', 'Facility', 'Pool water is not clean.', 'Resolved'),
(5, 208, '2023-08-05', 'Noise', 'Elevator noise.', 'Open'),
(6, 209, '2023-09-01', 'Maintenance', 'Plumbing issue in bathroom.', 'Resolved'),
(6, 210, '2023-09-10', 'Facility', 'Lights in hallway flickering.', 'Resolved'),
(7, 211, '2023-10-01', 'Noise', 'Neighbors playing loud music.', 'Open'),
(7, 212, '2023-10-05', 'Maintenance', 'Broken window in living room.', 'Resolved');
/* Insert Utility Bills */
INSERT INTO UTILITY_BILLING (Building_ID, Apartment_ID, UtilityType, MonthlyCost, DueDate) VALUES
(3, 203, 'Electricity', 160.00, '2023-06-01'),
(3, 204, 'Water', 80.00, '2023-06-01'),
(4, 205, 'Gas', 100.00, '2023-07-01'),
(4, 206, 'Electricity', 170.00, '2023-07-01'),
(5, 207, 'Water', 85.00, '2023-08-01'),
(5, 208, 'Gas', 95.00, '2023-08-01'),
(6, 209, 'Electricity', 150.00, '2023-09-01'),
(6, 210, 'Water', 90.00, '2023-09-01'),
(7, 211, 'Gas', 110.00, '2023-10-01'),
(7, 212, 'Electricity', 175.00, '2023-10-01');
/* Re-enable foreign key checks */
SET FOREIGN_KEY_CHECKS = 1;

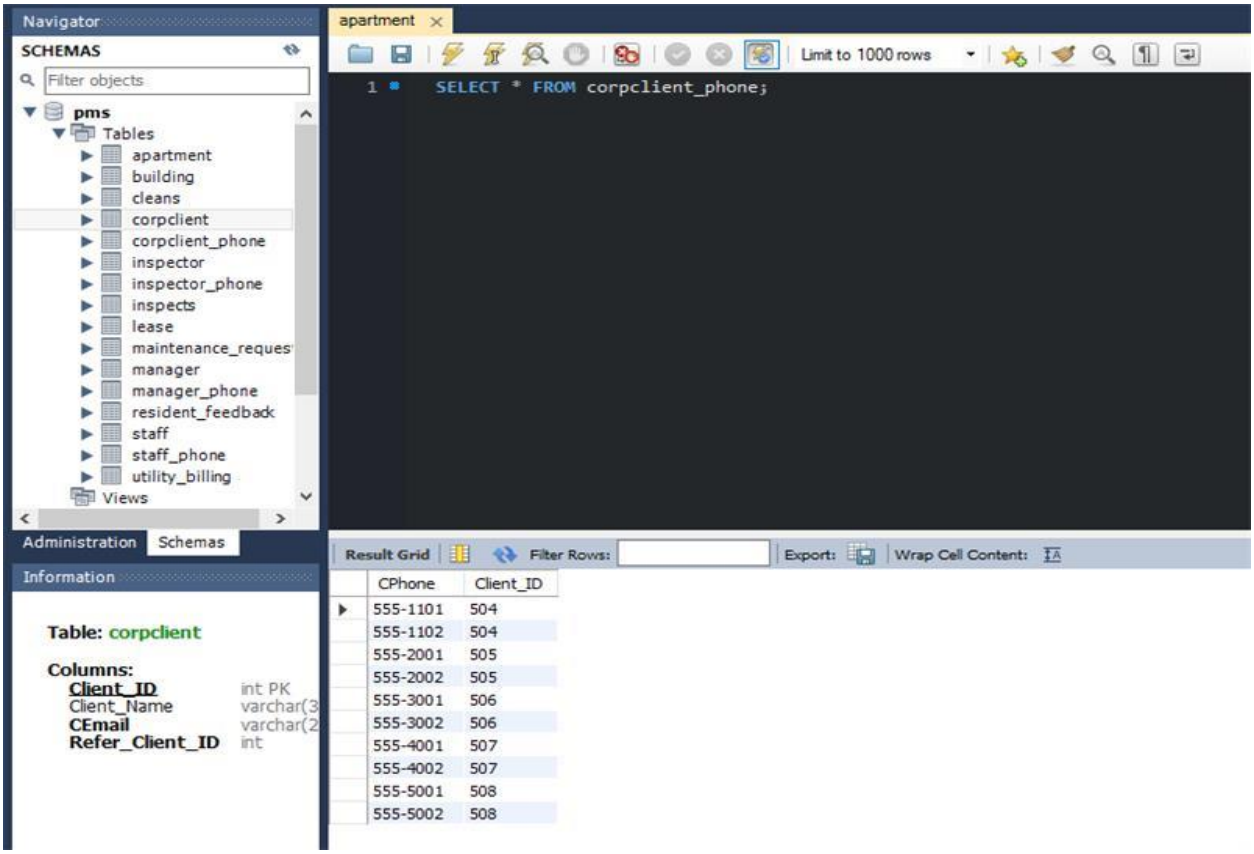
7.2 The SELECT * FROM TABLE statement in SQL would return all columns and all rows from the specified table.

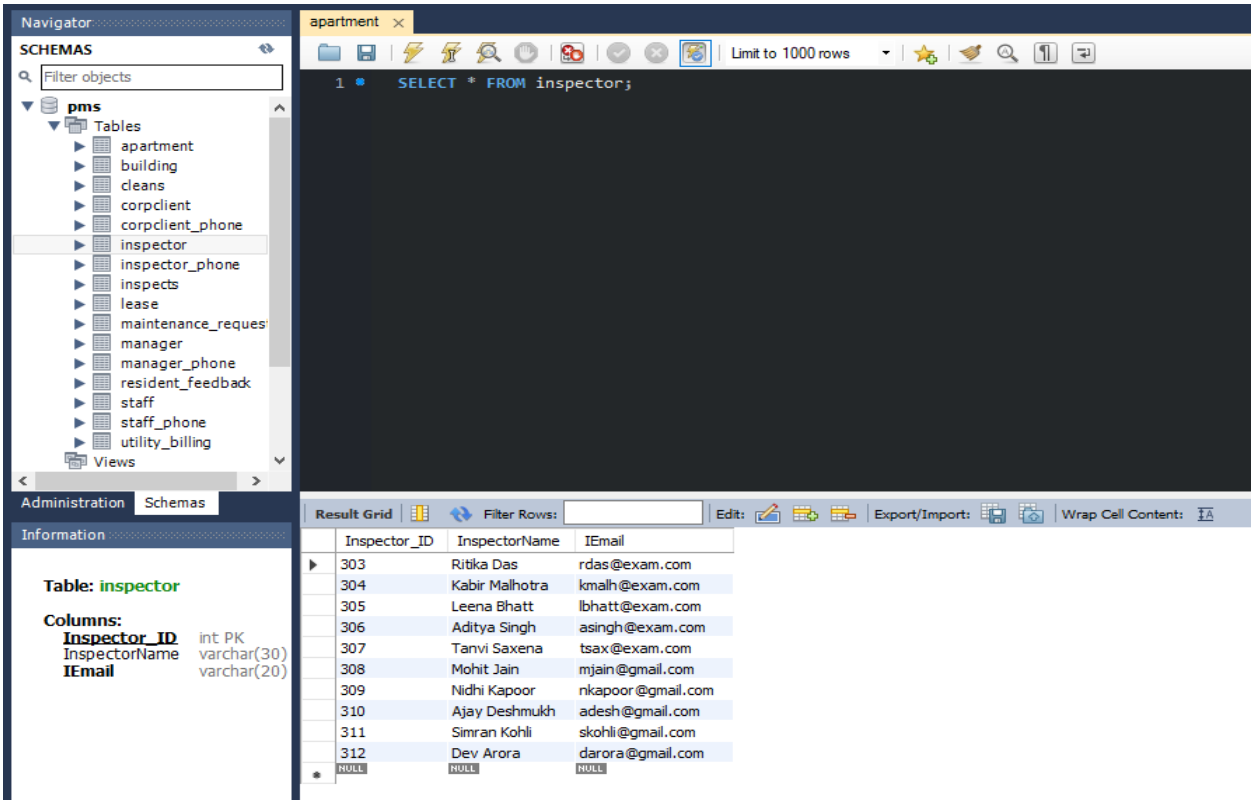
This following snippets show the details of all the tables from the query:

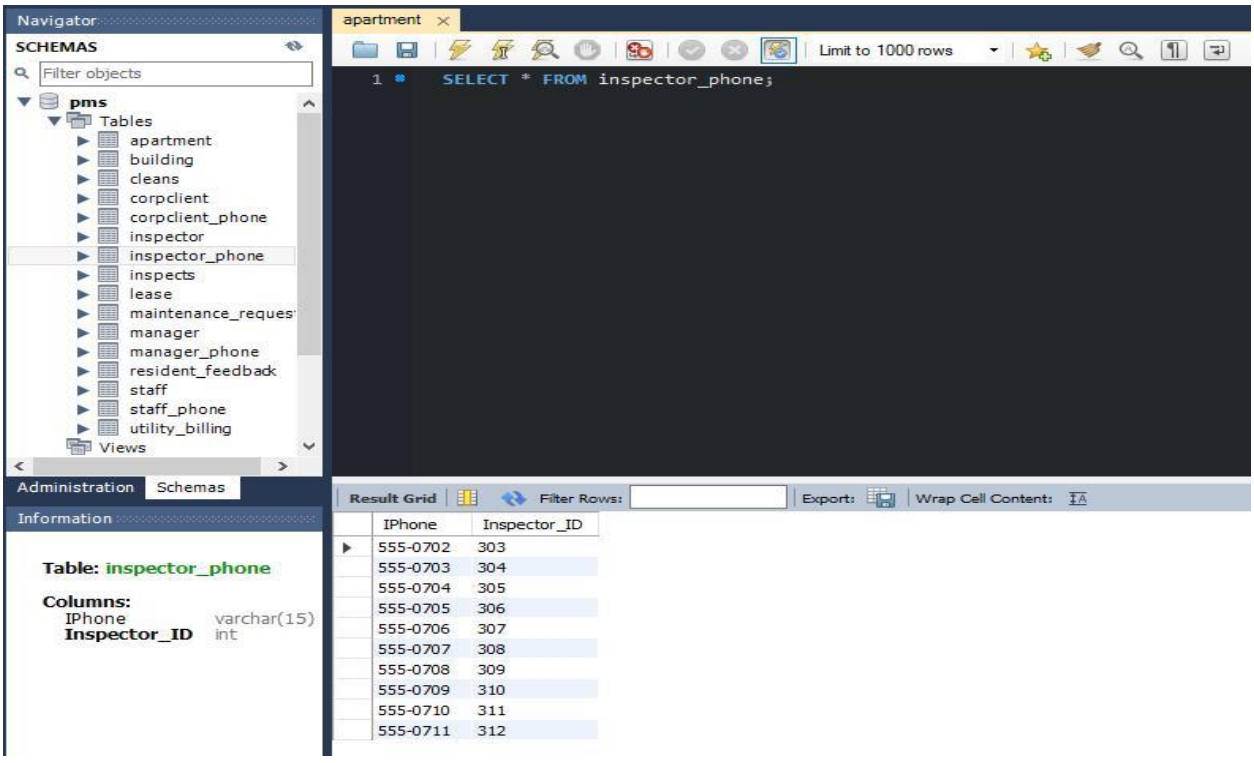


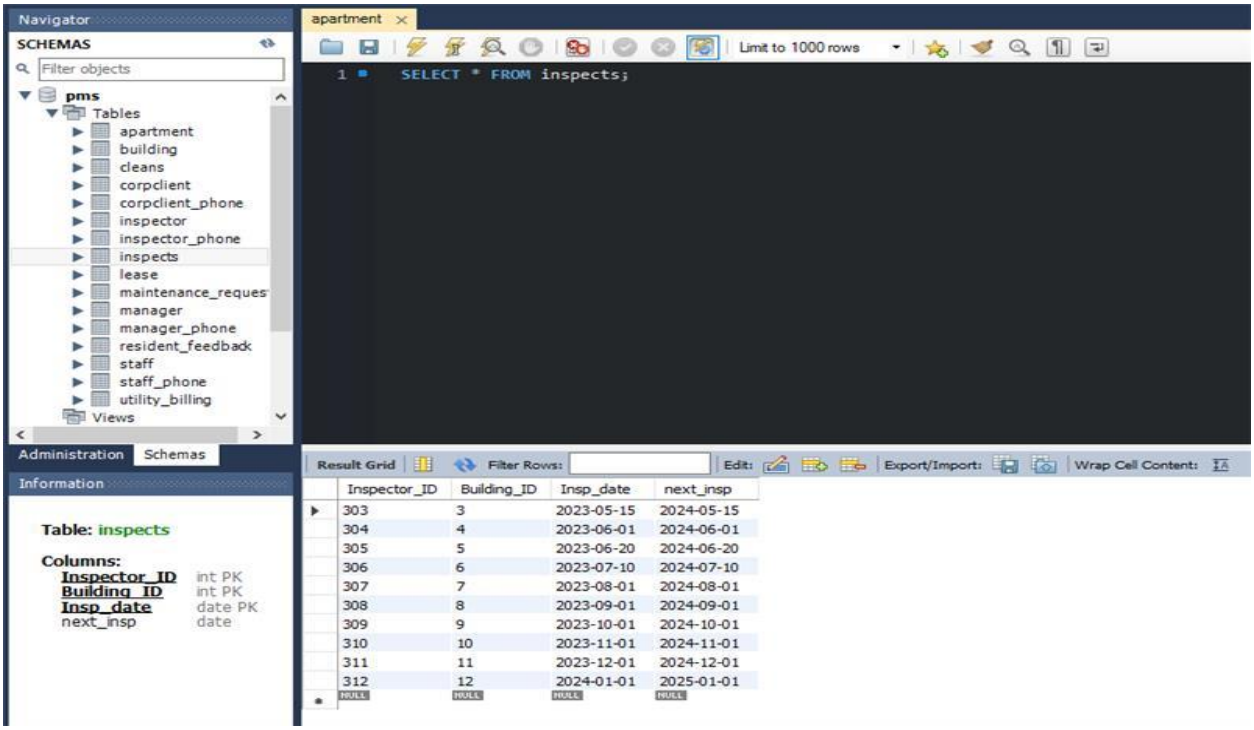


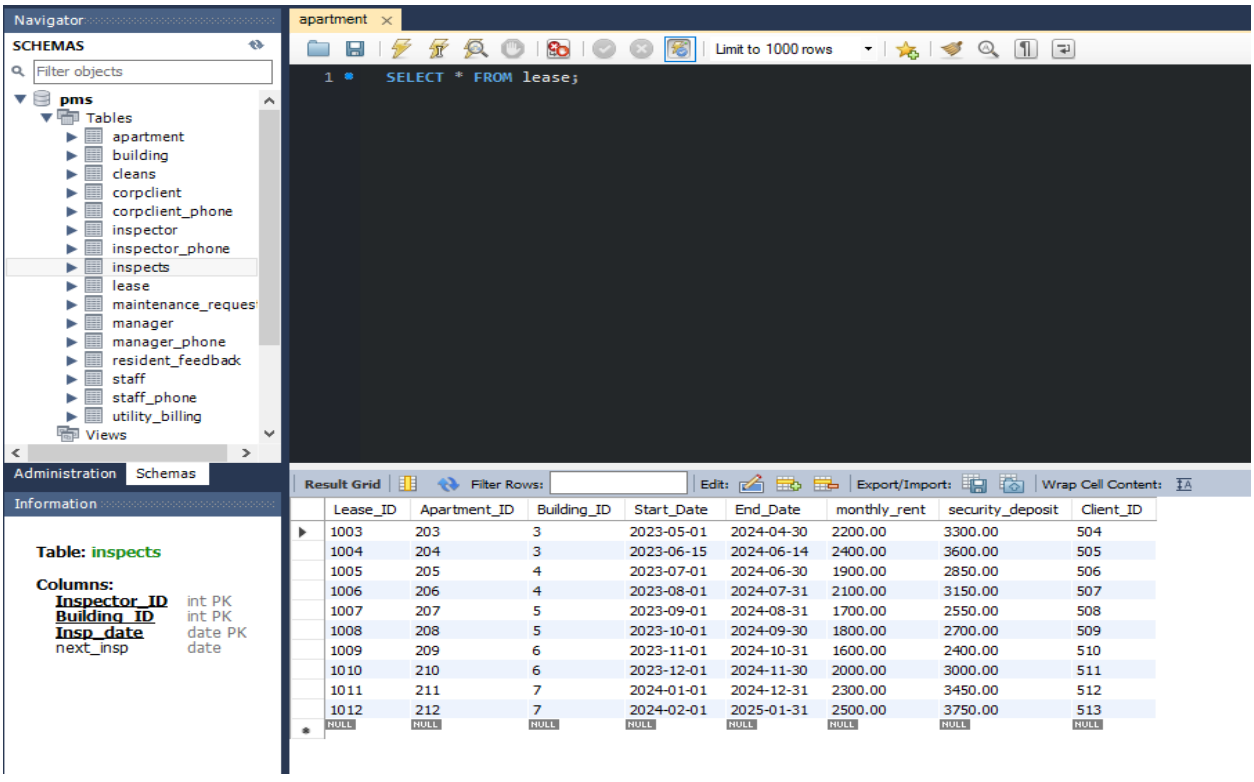

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: corpclient
Columns:
Client_ID int PK
Client_Name varchar(3
CEmail varchar(2
Refer_Client_ID int
apartment
Limit to 1000 rows
1 SELECT * FROM corpclient;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Client_ID | Client_Name | CEmail | Refer_Client_ID
504 | Innovative Solutions Ltd | innovate@gmail.com | NULL
505 | Eco Friendly Enterprises | eco@gmail.com | 504
506 | Smart Tech Corp | smart@gmail.com | 505
507 | Global Logistics Inc | log@gmail.com | 506
508 | Creative Designs Studio | desi@gmail.com | 507
509 | Health Plus Clinic | health@gmail.com | 508
510 | EduTech Solutions | edutech@gmail.com | 509
511 | Green Energy Co | green@gmail.com | 510
512 | Fast Delivery Services | del@gmail.com | 511
513 | Bright Future School | sc@gmail.com | 512

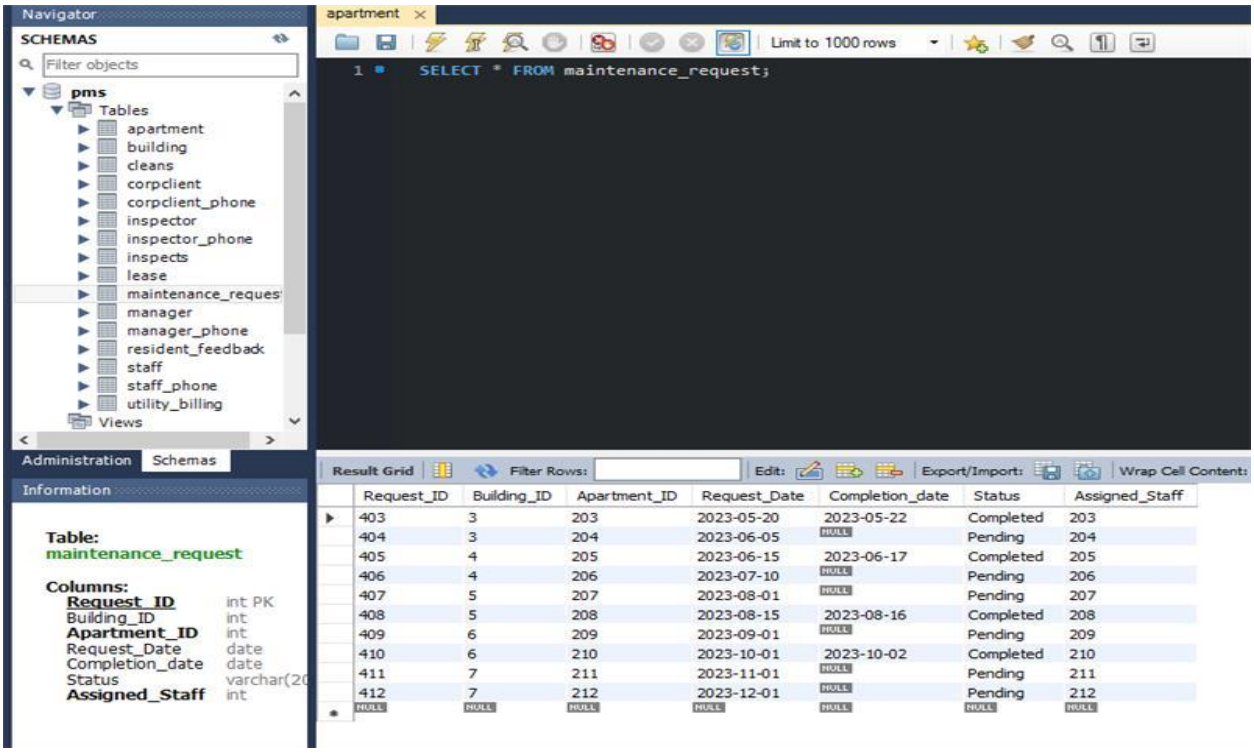

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: corpclient
Columns:
Client_ID int PK
Client_Name varchar(3
CEmail varchar(2
Refer_Client_ID int
apartment
Limit to 1000 rows
1 SELECT * FROM corpclient_phone;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
CPhone | Client_ID
555-1101 | 504
555-1102 | 504
555-2001 | 505
555-2002 | 505
555-3001 | 506
555-3002 | 506
555-4001 | 507
555-4002 | 507
555-5001 | 508
555-5002 | 508

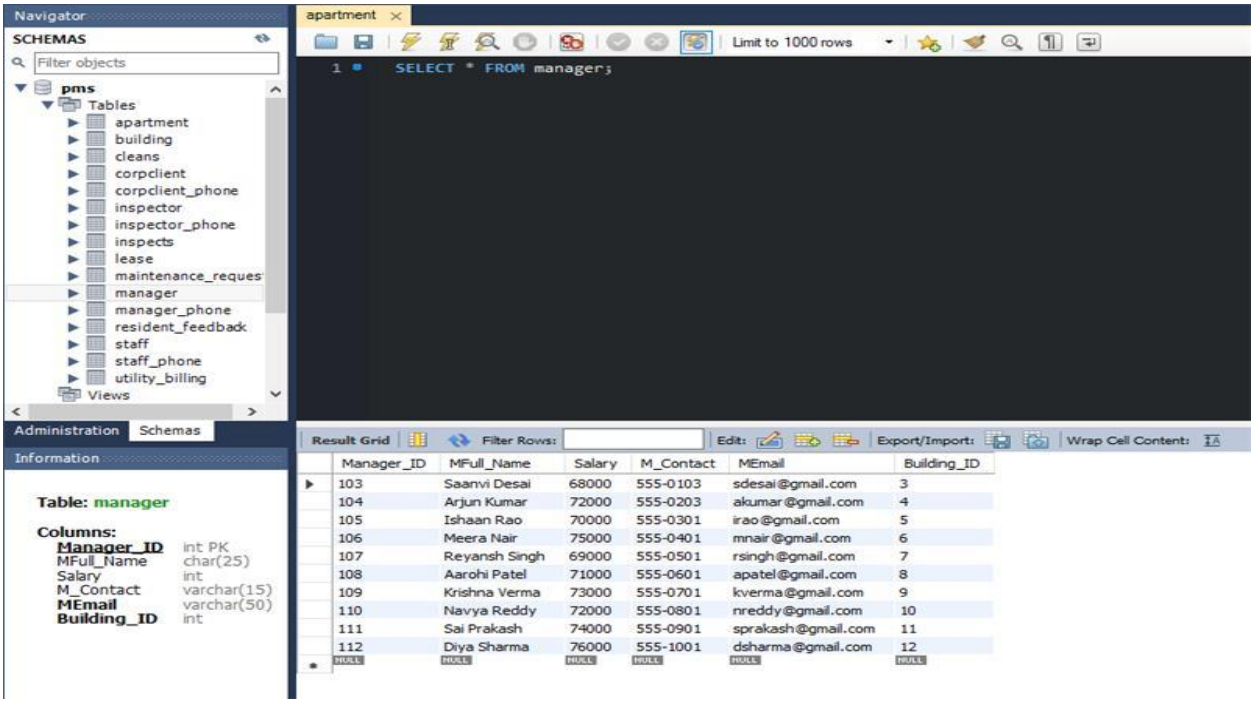

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: inspector
Columns:
Inspector_ID int PK
InspectorName varchar(30)
IEmail varchar(20)
apartment
Limit to 1000 rows
1 SELECT * FROM inspector;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Inspector_ID InspectorName IEmail
303 Ritika Das rdas@exam.com
304 Kabir Malhotra kmalh@exam.com
305 Leena Bhatt lbhatt@exam.com
306 Aditya Singh asingh@exam.com
307 Tanvi Saxena tsax@exam.com
308 Mohit Jain mjain@gmail.com
309 Nidhi Kapoor nkapoor@gmail.com
310 Ajay Deshmukh adesh@gmail.com
311 Simran Kohli skohli@gmail.com
312 Dev Arora darora@gmail.com

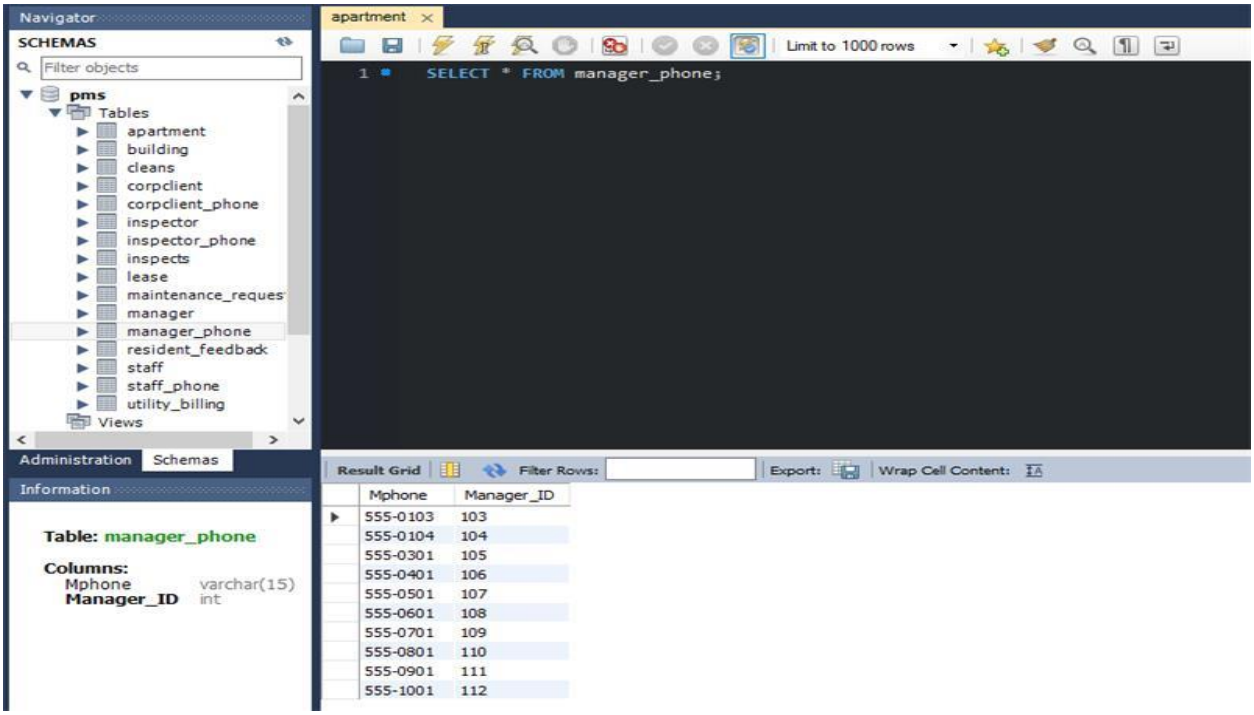

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: inspector_phone
Columns:
IPhone varchar(15)
Inspector_ID int
apartment
Limit to 1000 rows
1 SELECT * FROM inspector_phone;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
IPhone Inspector_ID
555-0702 303
555-0703 304
555-0704 305
555-0705 306
555-0706 307
555-0707 308
555-0708 309
555-0709 310
555-0710 311
555-0711 312

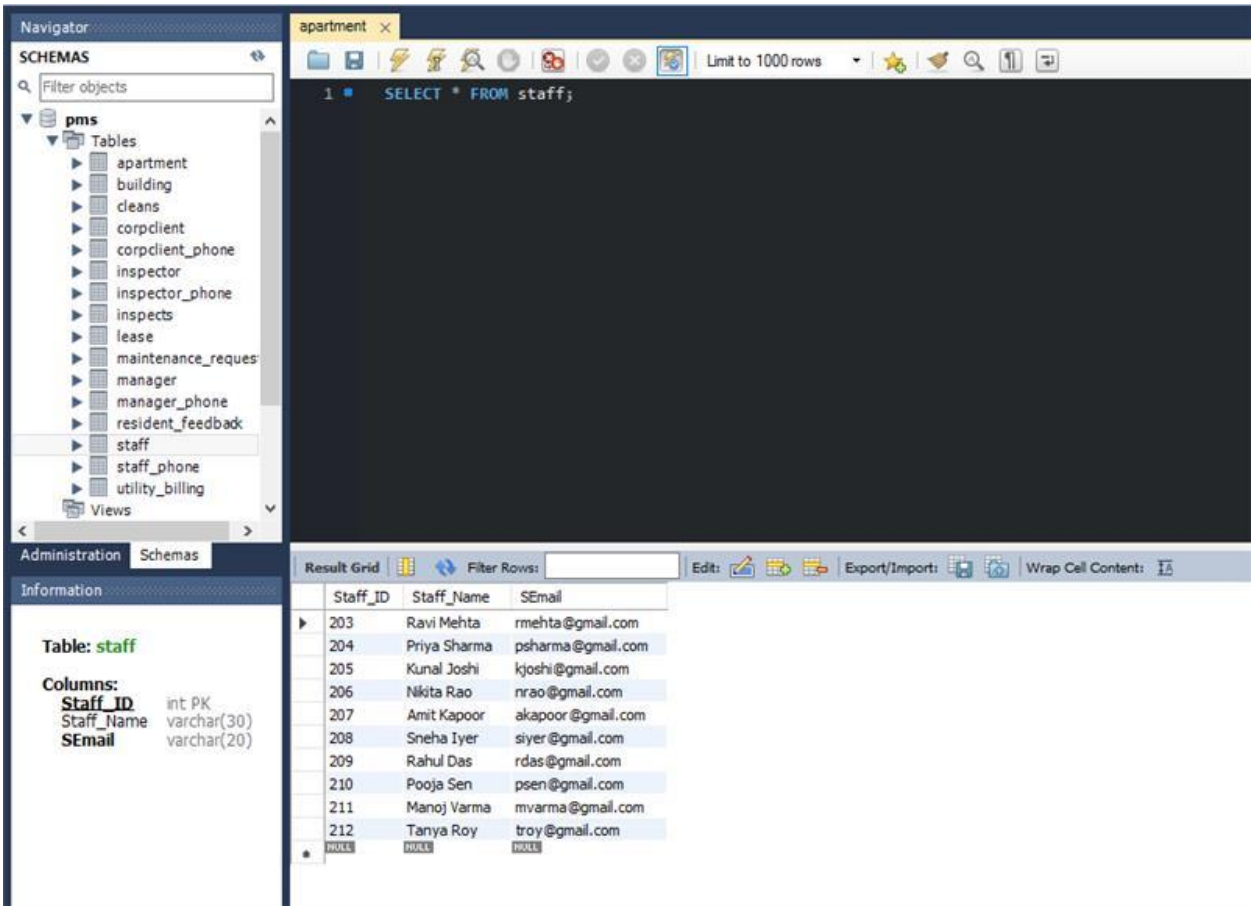

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: inspects
Columns:
Inspector_ID int PK
Building_ID int PK
Insp_date date PK
next_insp date
apartment
Limit to 1000 rows
1 SELECT * FROM inspects;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Inspector_ID Building_ID Insp_date next_insp
303 3 2023-05-15 2024-05-15
304 4 2023-06-01 2024-06-01
305 5 2023-06-20 2024-06-20
306 6 2023-07-10 2024-07-10
307 7 2023-08-01 2024-08-01
308 8 2023-09-01 2024-09-01
309 9 2023-10-01 2024-10-01
310 10 2023-11-01 2024-11-01
311 11 2023-12-01 2024-12-01
312 12 2024-01-01 2025-01-01
NULL NULL NULL NULL

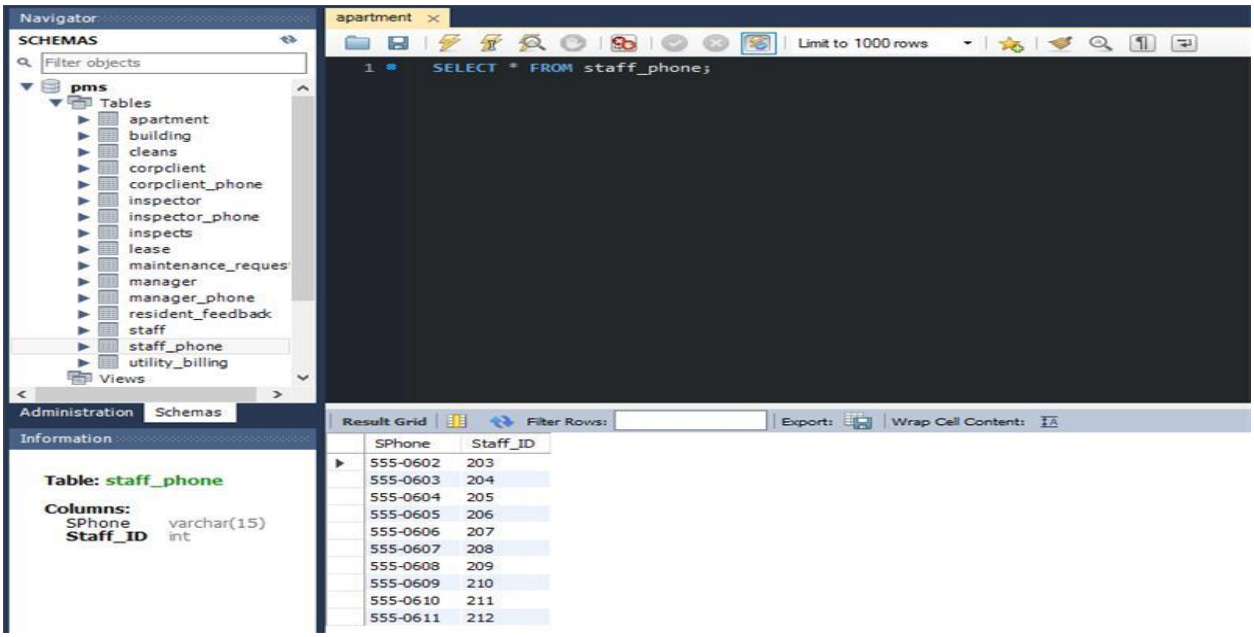

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: inspects
Columns:
Inspector_ID int PK
Building_ID int PK
Insp_date date PK
next_insp date
apartment
Limit to 1000 rows
1 SELECT * FROM lease;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Lease_ID Apartment_ID Building_ID Start_Date End_Date monthly_rent security_deposit Client_ID
1003 203 3 2023-05-01 2024-04-30 2200.00 3300.00 504
1004 204 3 2023-06-15 2024-06-14 2400.00 3600.00 505
1005 205 4 2023-07-01 2024-06-30 1900.00 2850.00 506
1006 206 4 2023-08-01 2024-07-31 2100.00 3150.00 507
1007 207 5 2023-09-01 2024-08-31 1700.00 2550.00 508
1008 208 5 2023-10-01 2024-09-30 1800.00 2700.00 509
1009 209 6 2023-11-01 2024-10-31 1600.00 2400.00 510
1010 210 6 2023-12-01 2024-11-30 2000.00 3000.00 511
1011 211 7 2024-01-01 2024-12-31 2300.00 3450.00 512
1012 212 7 2024-02-01 2025-01-31 2500.00 3750.00 513
NULL NULL NULL NULL NULL NULL NULL NULL


Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table:
maintenance_request
Columns:
Request_ID int PK
Building_ID int
Apartment_ID int
Request_Date date
Completion_date date
Status varchar(20
Assigned_Staff int
apartment
Limit to 1000 rows
1 SELECT * FROM maintenance_request;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Request_ID Building_ID Apartment_ID Request_Date Completion_date Status Assigned_Staff
403 3 203 2023-05-20 2023-05-22 Completed 203
404 3 204 2023-06-05 NULL Pending 204
405 4 205 2023-06-15 2023-06-17 Completed 205
406 4 206 2023-07-10 NULL Pending 206
407 5 207 2023-08-01 NULL Pending 207
408 5 208 2023-08-15 2023-08-16 Completed 208
409 6 209 2023-09-01 NULL Pending 209
410 6 210 2023-10-01 2023-10-02 Completed 210
411 7 211 2023-11-01 NULL Pending 211
412 7 212 2023-12-01 NULL Pending 212


Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: manager
Columns:
Manager_ID int PK
MFull_Name char(25)
Salary int
M_Contact varchar(15)
MEmail varchar(50)
Building_ID int
apartment
Limit to 1000 rows
1 SELECT * FROM manager;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Manager_ID MFull_Name Salary M_Contact MEmail Building_ID
103 Saanvi Desai 68000 555-0103 sdesai@gmail.com 3
104 Arjun Kumar 72000 555-0203 akumar@gmail.com 4
105 Ishaan Rao 70000 555-0301 irao@gmail.com 5
106 Meera Nair 75000 555-0401 mnair@gmail.com 6
107 Reyansh Singh 69000 555-0501 rsingh@gmail.com 7
108 Aarohi Patel 71000 555-0601 apatel@gmail.com 8
109 Krishna Verma 73000 555-0701 kverma@gmail.com 9
110 Navya Reddy 72000 555-0801 nreddy@gmail.com 10
111 Sai Prakash 74000 555-0901 sprakash@gmail.com 11
112 Diya Sharma 76000 555-1001 dsharma@gmail.com 12

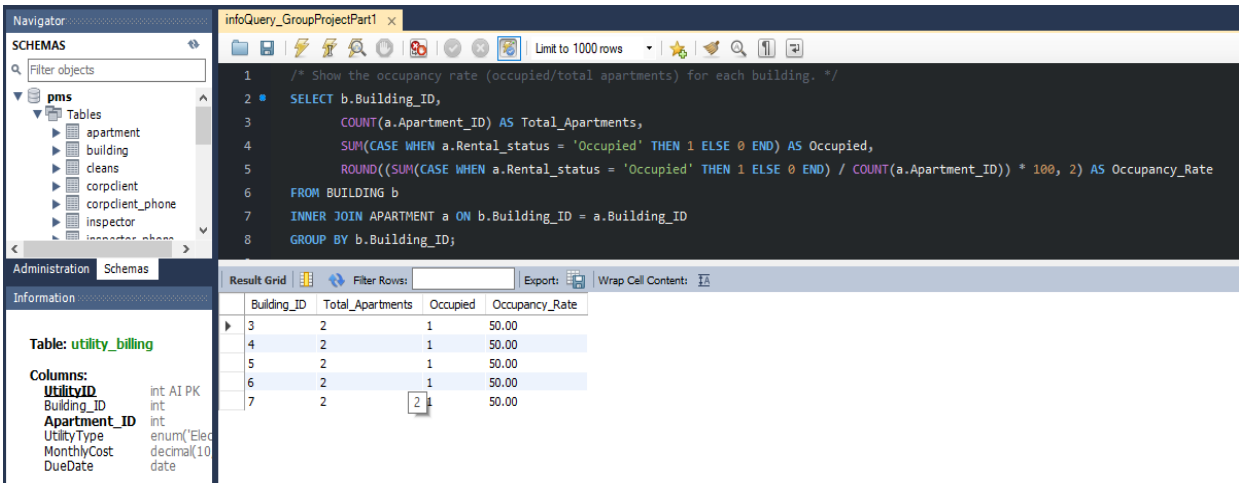

Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: manager_phone
Columns:
Mphone varchar(15)
Manager_ID int
apartment
Limit to 1000 rows
1 • SELECT * FROM manager_phone;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Mphone
Manager_ID
555-0103 103
555-0104 104
555-0301 105
555-0401 106
555-0501 107
555-0601 108
555-0701 109
555-0801 110
555-0901 111
555-1001 112


Navigator
SCHEMAS
Filter objects
pms
Tables
apartment
building
cleans
corpclient
corpclient_phone
inspector
inspector_phone
inspects
lease
maintenance_reques
manager
manager_phone
resident_feedback
staff
staff_phone
utility_billing
Views
Administration
Schemas
Information
Table: staff
Columns:
Staff_ID int PK
Staff_Name varchar(30)
SEmail varchar(20)
apartment
Limit to 1000 rows
1 • SELECT * FROM staff;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Staff_ID
Staff_Name
SEmail
203 Ravi Mehta rmehta@gmail.com
204 Priya Sharma psharma@gmail.com
205 Kunal Joshi kjoshi@gmail.com
206 Nikita Rao nrao@gmail.com
207 Amit Kapoor akapoor@gmail.com
208 Sneha Iyer siyer@gmail.com
209 Rahul Das rdas@gmail.com
210 Pooja Sen psen@gmail.com
211 Manoj Varma mvarma@gmail.com
212 Tanya Roy troy@gmail.com

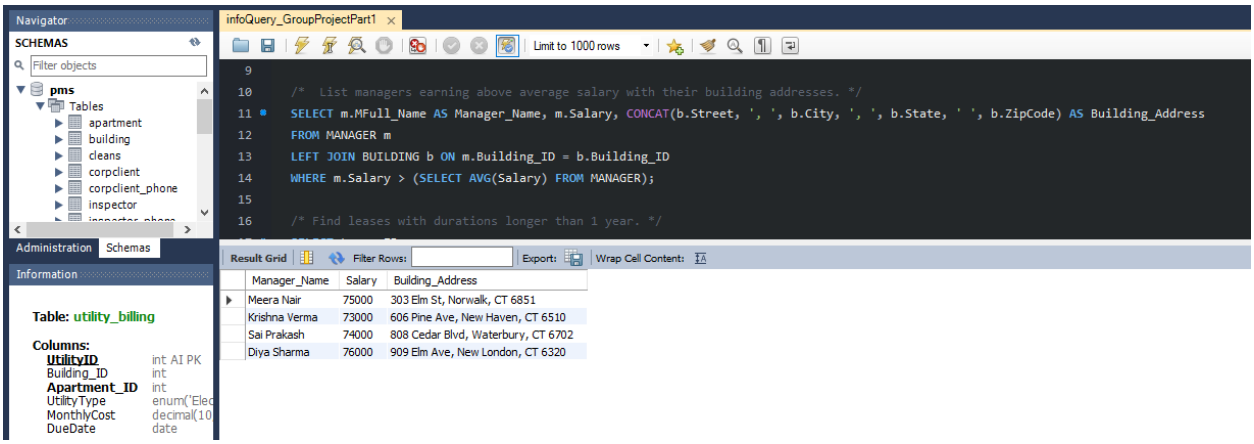


8. **(30 points) Create 10 business questions and provide the SQL codes and results. Submit the questions, SQL codes and the screenshots of the results. Use different types of joins, aggregation function, sub query, CHECK, stored procedure, trigger at least once.**

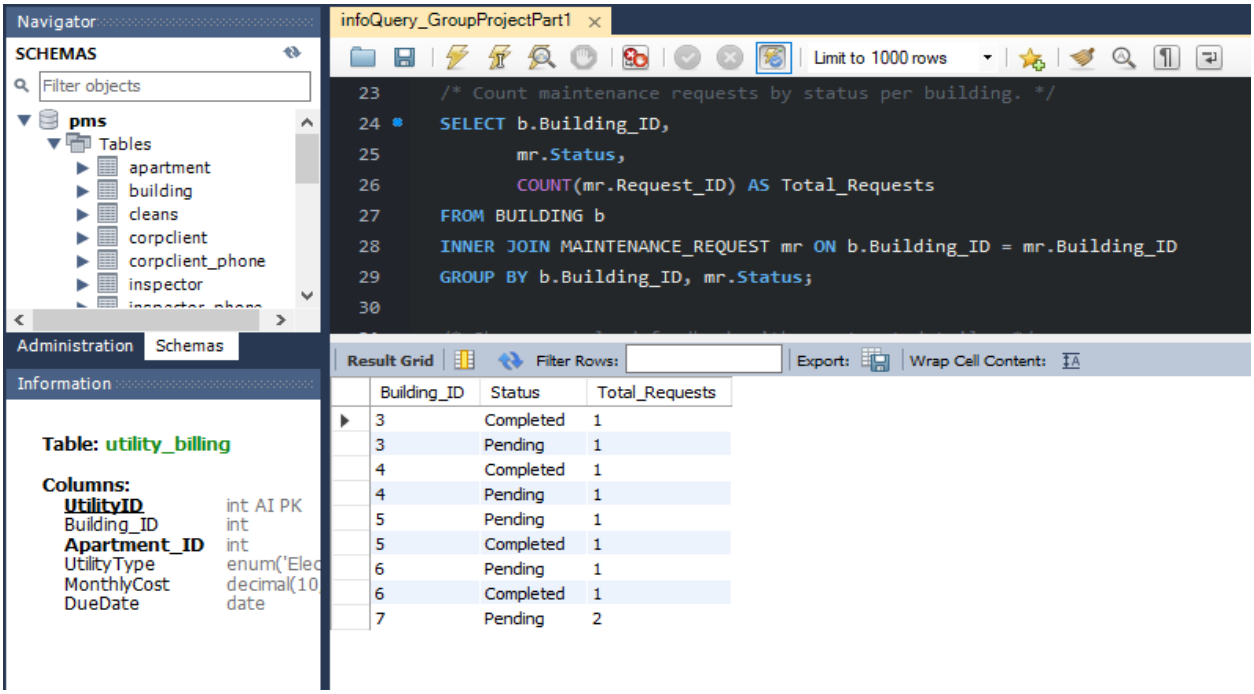
8.1 Show the occupancy rate (occupied/total apartments) for each building.



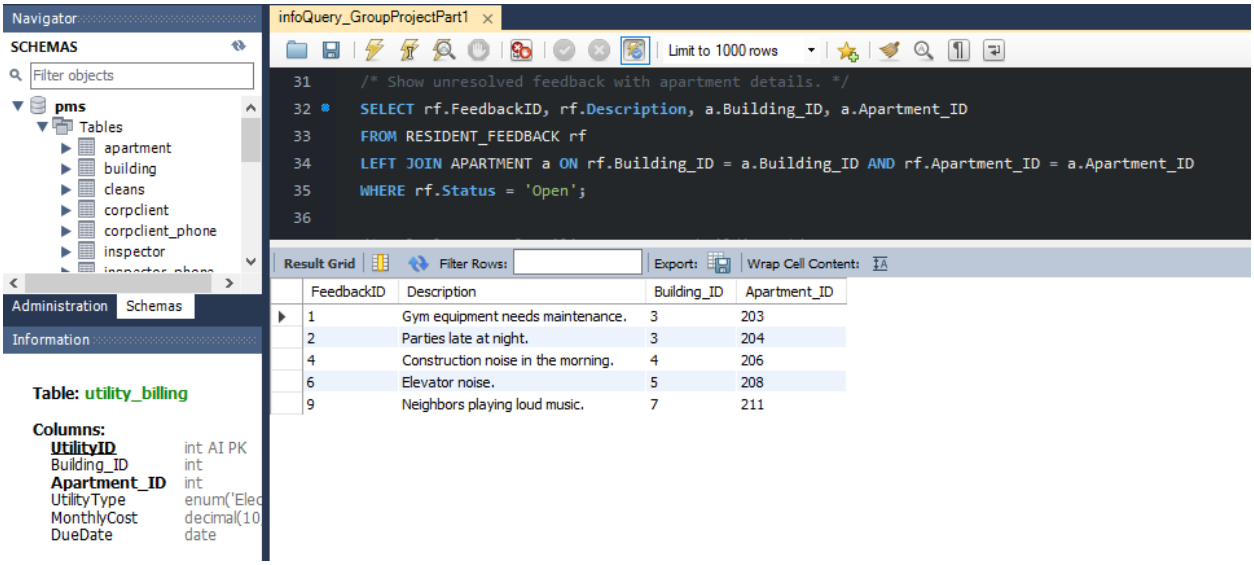
8.2 List managers earning above average salary with their building addresses.



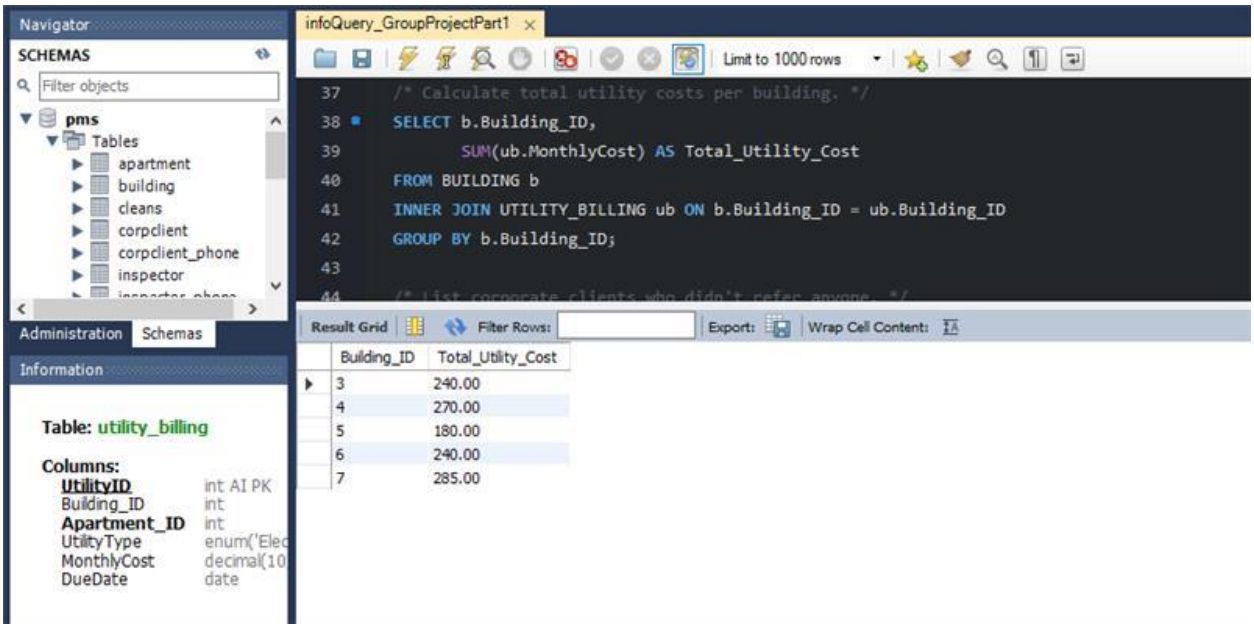
8.3 Count maintenance requests by status per building.



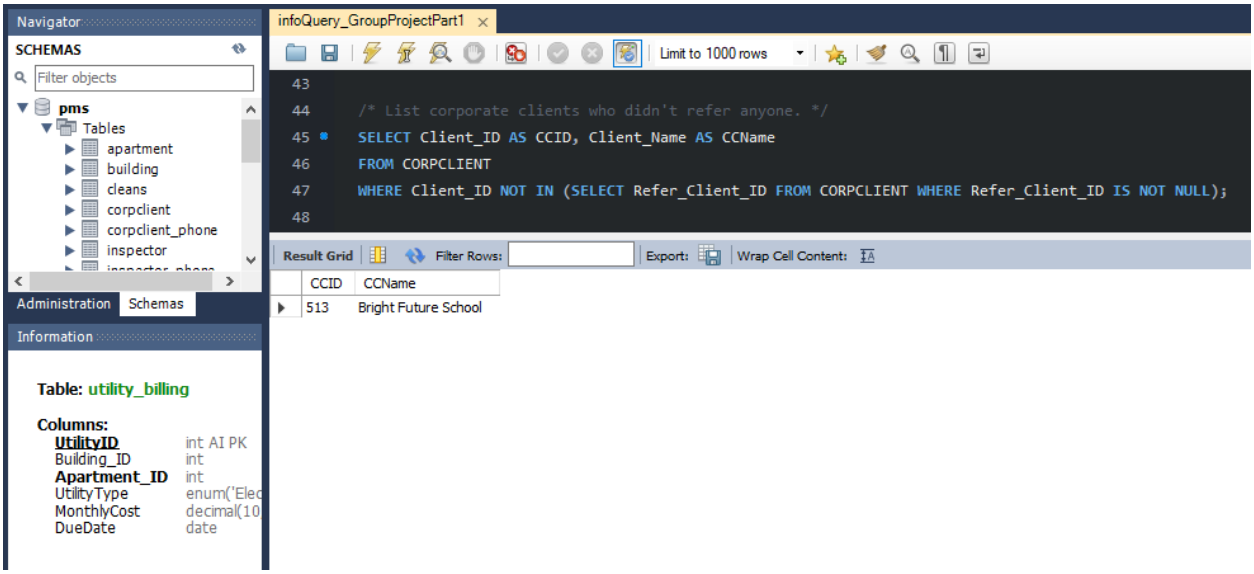
8.4 Show unresolved feedback with apartment details.



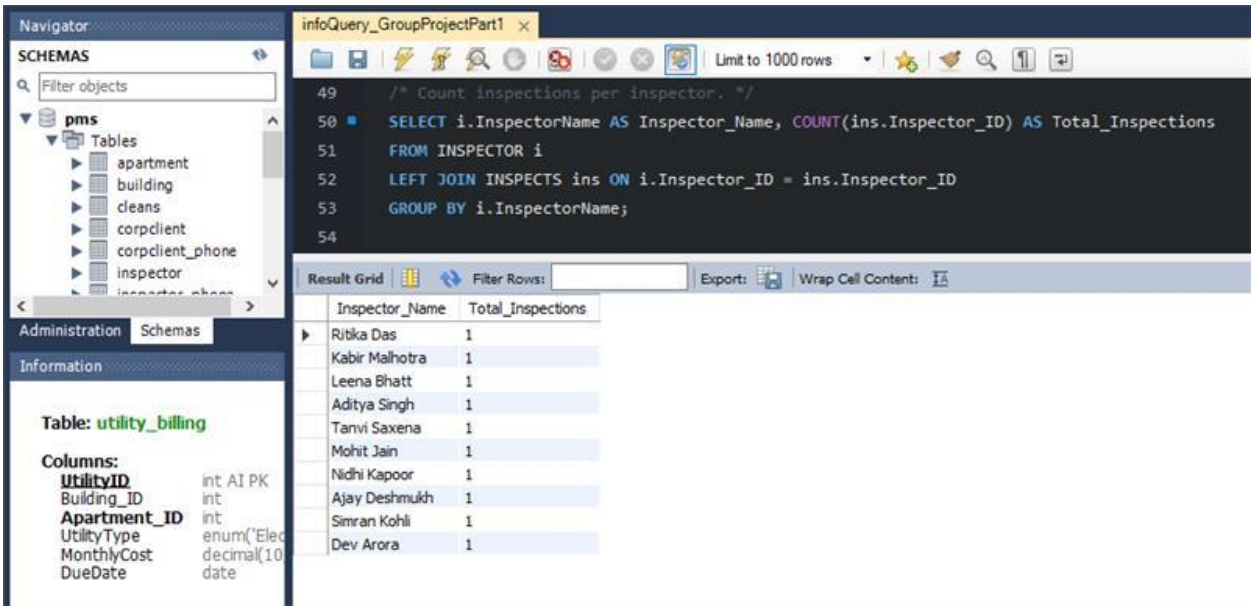
8.5 Calculate total utility costs per building.



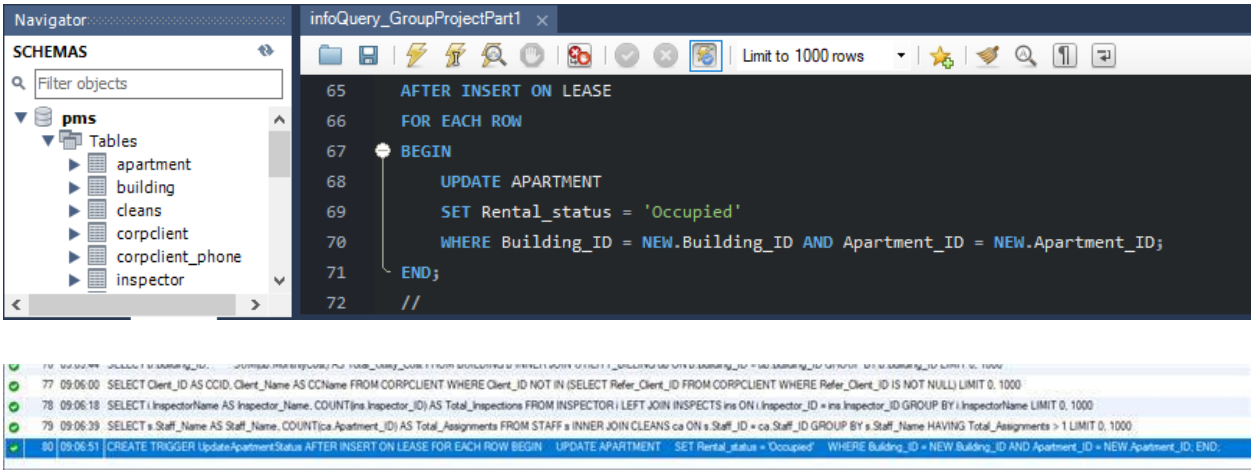
8.6 List corporate clients who didn't refer anyone.



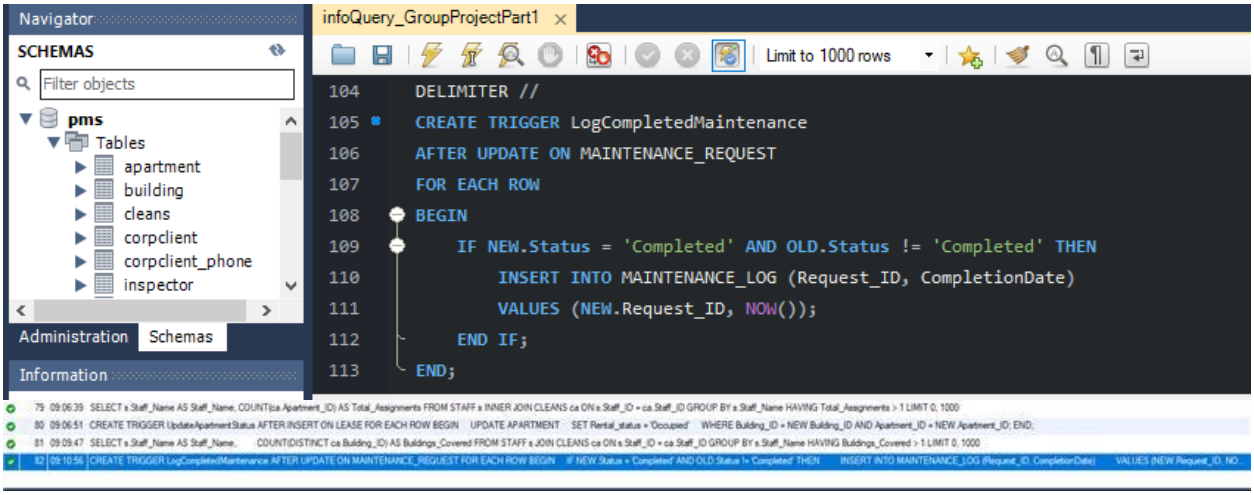
8.7 Count inspections per inspector.



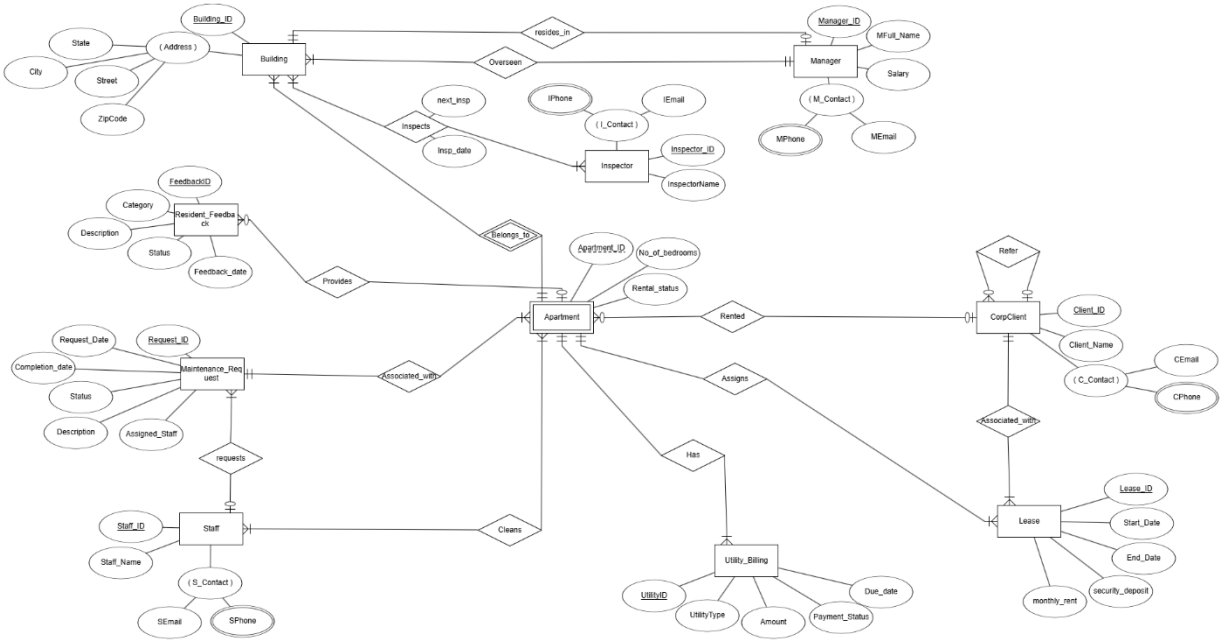
8.8 Create a trigger to mark apartments as occupied when leased.



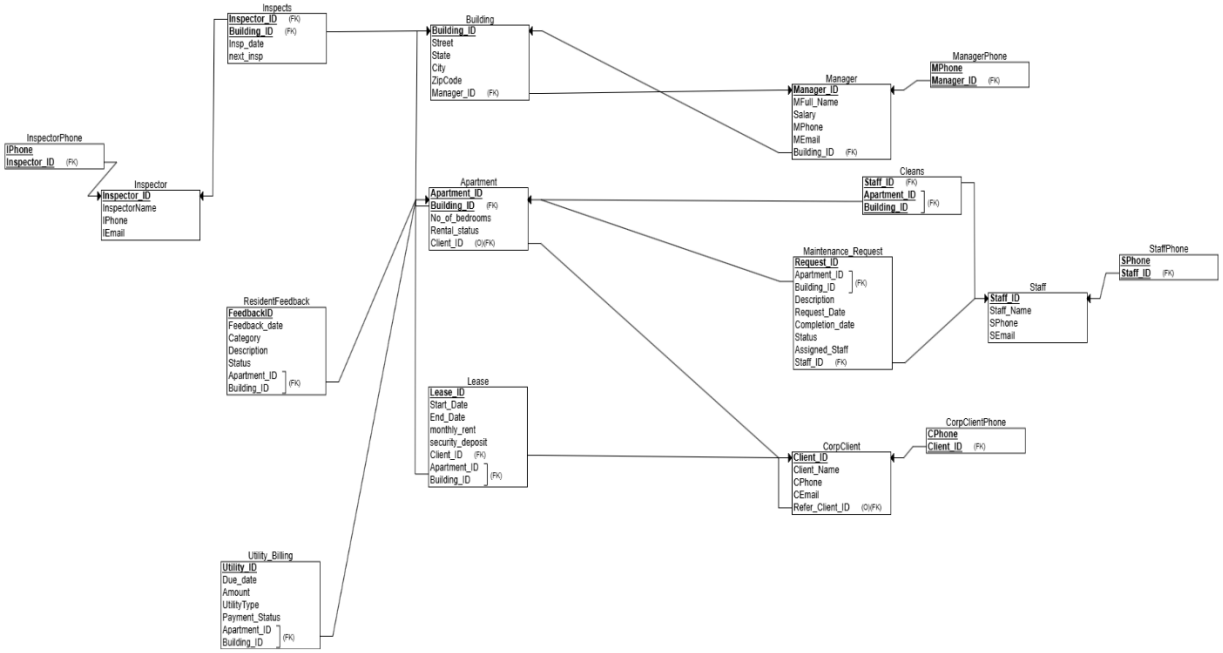
8.9 Automatically log completed maintenance requests



9. Entity-Relationship: PROPERTY MANAGEMENT SYSTEM: 2 NEW ENTITIES



9.1 Relational-Schema: PROPERTY MANAGEMENT SYSTEM: 2 NEW ENTITIES

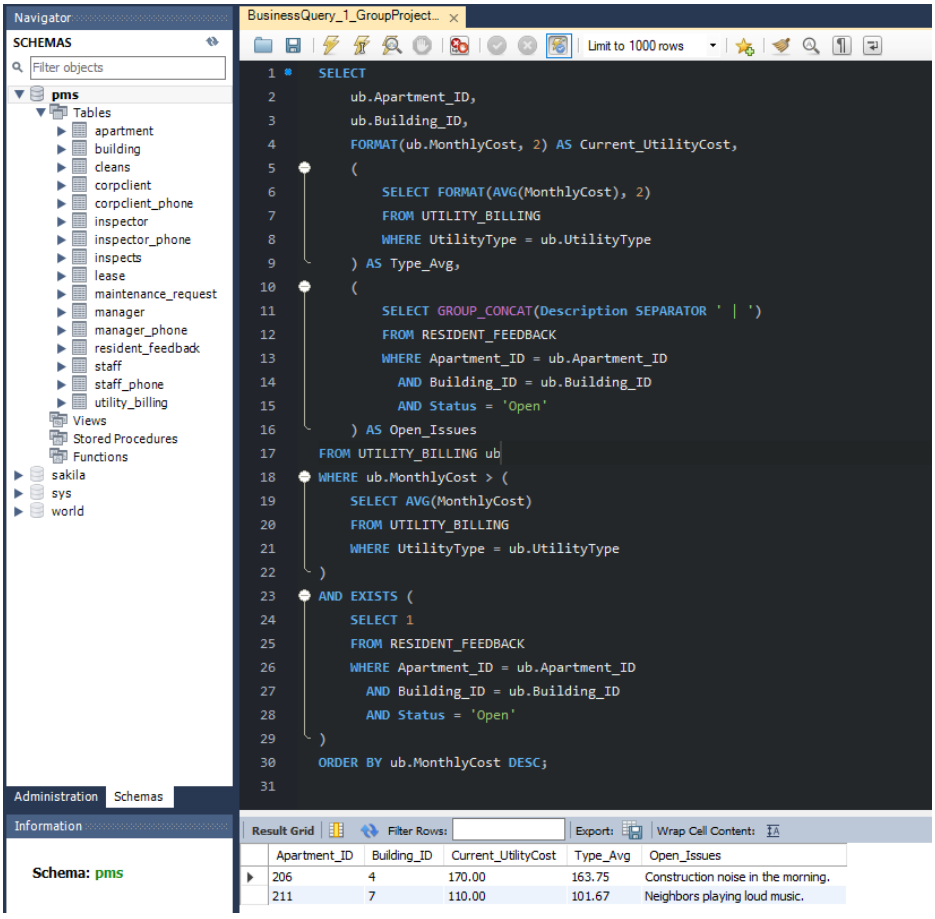


9.2 DATA-DICTIONARY: PROPERTY MANAGEMENT SYSTEM: 2 NEW ENTITIES

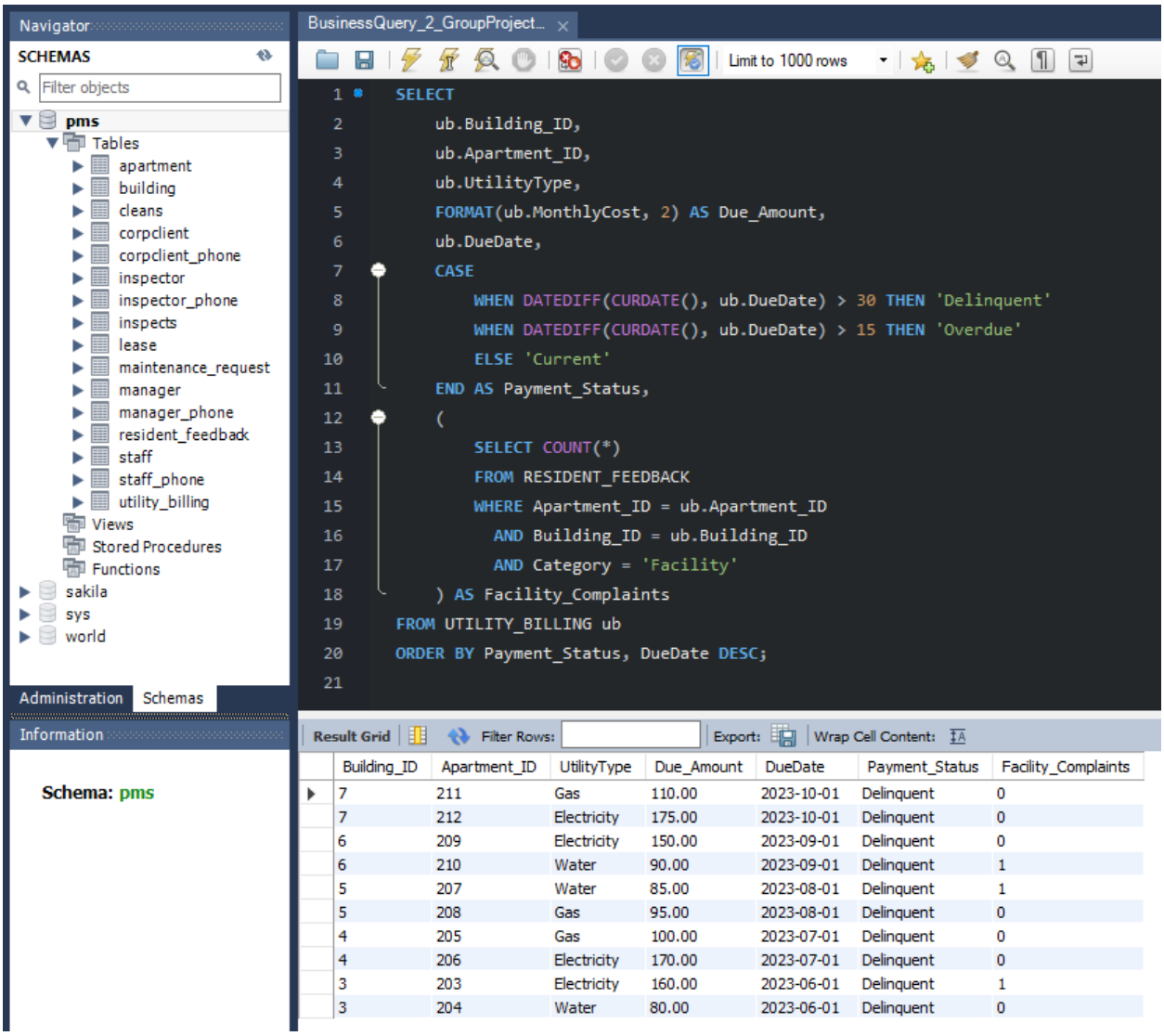
Entity:	RESIDENT FEEDBACK TABLE	Track utility billing of apartments					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
FeedbackID	Unique feedback ID	INT	Auto-increment	-	YES	YES	PK
BuildingID	Building ID	INT	Links to Apartment	-	YES	NO	-
AptNo	Apartment number	INT	Links to Apartment	-	YES	NO	-
FeedbackDate	Date of feedback submission	DATE	Format: YYYY-MM-DD	-	YES	NO	-
Category	Feedback category	ENUM	MaINTenance, Noise, Facility	-	YES	NO	-
Description	Feedback details	TEXT		-	YES	NO	-
Status	Feedback resolution status	VARCHAR(10)	CHECK: Open/Resolved	-	YES	NO	-
Entity:	**UTILITY BILLING**	Track feedbacks from residents					
Field Name	**Description**	**Type**	**Specifications**	**Default**	**Required**	**Unique**	**Key(s)**
UtilityID	Unique utility bill ID	INT	Auto-increment	-	YES	YES	PK
BuildingID	Building ID	INT	Links to Apartment	-	YES	NO	-
AptNo	Apartment number	INT	Links to Apartment	-	YES	NO	-
UtilityType	Type of utility	ENUM	Electricity, Water, Gas	-	YES	NO	-
MonthlyCost	Monthly utility cost	DECIMAL(10,2)	Positive value (e.g., 150.00)	-	YES	NO	-
DueDate	Payment due date	DATE	Format: YYYY-MM-DD	-	YES	NO	-

9.3 BUSINESS QUERY: 2 NEW ENTITIES – RESIDENT FEEDBACK AND UTILITY BILLING

9.3.1 High Utility Cost Apartments with Pending Feedback

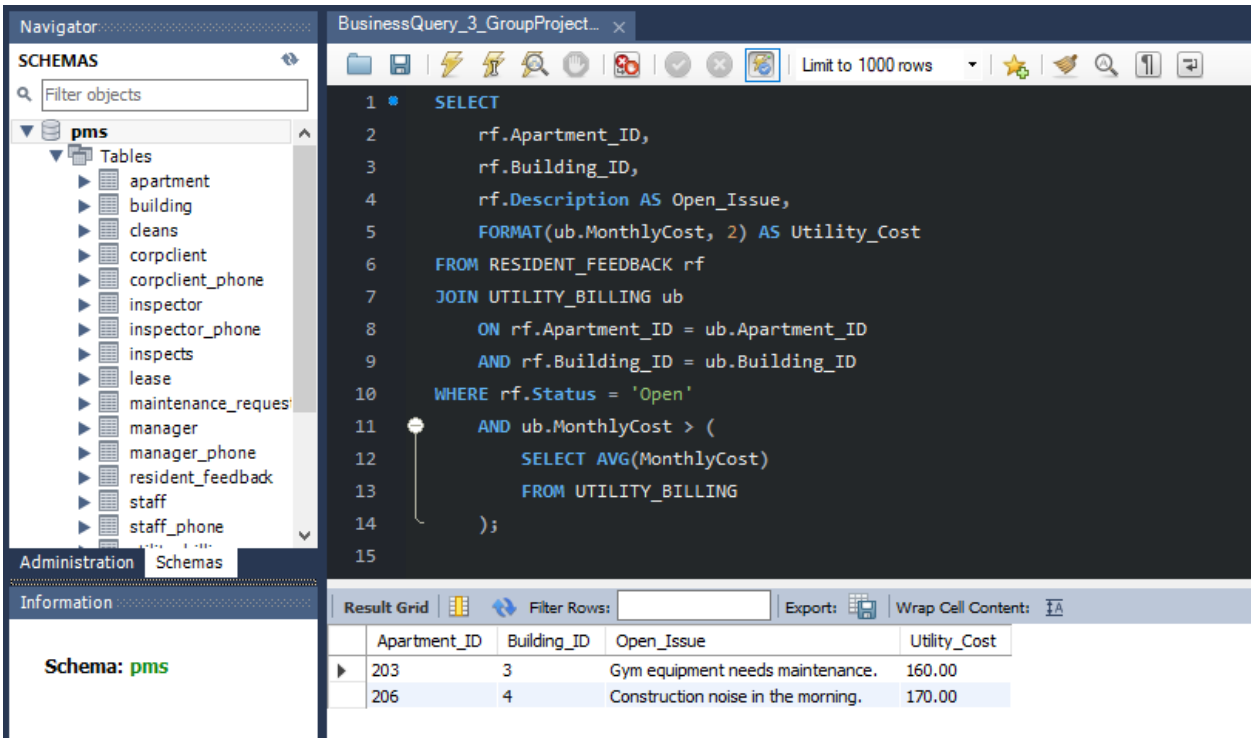


9.3.2 Utility payment compliance check

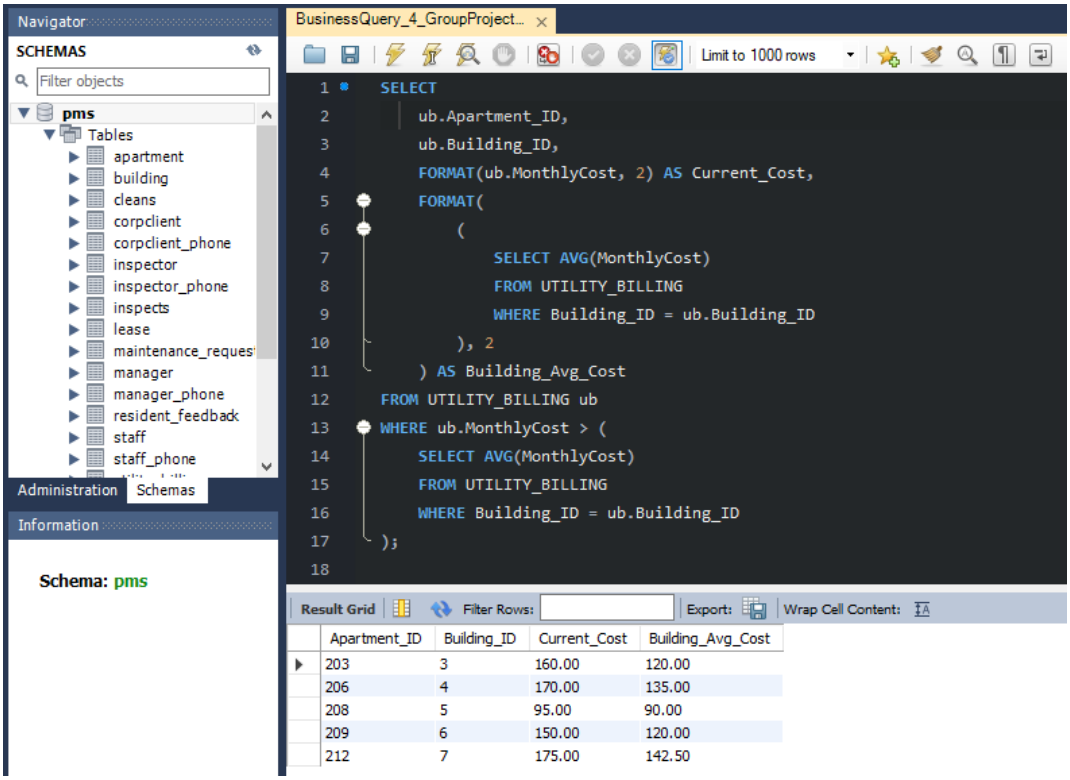


Building_ID	Apartment_ID	UtilityType	Due_Amount	DueDate	Payment_Status	Facility_Complaints
7	211	Gas	110.00	2023-10-01	Delinquent	0
7	212	Electricity	175.00	2023-10-01	Delinquent	0
6	209	Electricity	150.00	2023-09-01	Delinquent	0
6	210	Water	90.00	2023-09-01	Delinquent	1
5	207	Water	85.00	2023-08-01	Delinquent	1
5	208	Gas	95.00	2023-08-01	Delinquent	0
4	205	Gas	100.00	2023-07-01	Delinquent	0
4	206	Electricity	170.00	2023-07-01	Delinquent	0
3	203	Electricity	160.00	2023-06-01	Delinquent	1
3	204	Water	80.00	2023-06-01	Delinquent	0

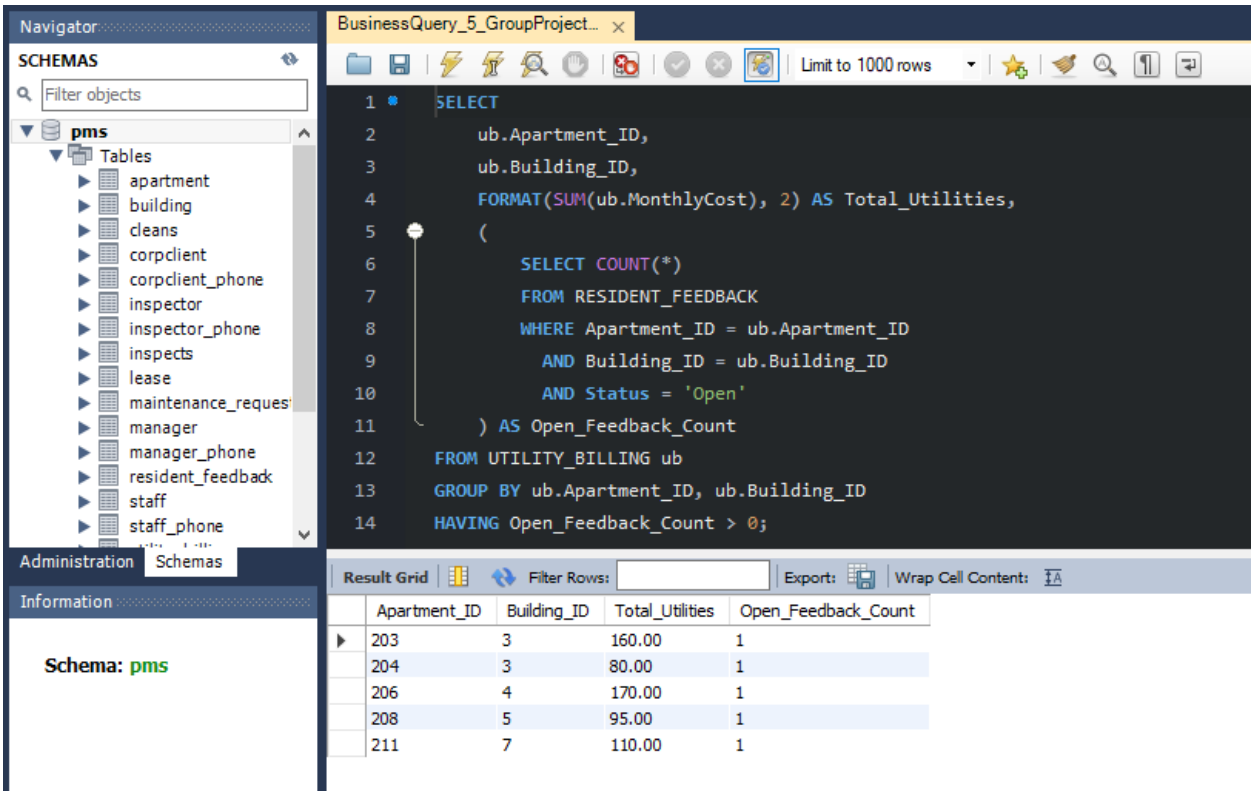
9.3.3 Apartments with Unresolved Feedback & High Utility Costs



9.3.4 Utility Costs vs Building Average



9.3.5 Open Feedback Count with Total Utility Cost



This project successfully addressed the property management company's operational and analytical needs by modernizing its database infrastructure.

Part 1: Operational Database Upgrade

1. **Enhanced Functionality:**
    - The upgraded database now efficiently tracks leases and maintenance requests, streamlining rental management and repair workflows.
    - New entities like Resident Feedback and Utility Billing were added to improve tenant satisfaction monitoring and cost tracking.
2. **Data Integrity & Efficiency:**
    - Triggers (e.g., auto-updating apartment vacancy status) and stored procedures (e.g., rent revenue calculation) reduced manual effort.
    - Referential integrity was enforced through foreign keys and constraints (e.g., CHECK clauses for valid statuses).
3. **Business Insights:**

    - Queries using joins, aggregations, and subqueries answered critical questions, such as identifying unoccupied apartments or analyzing referral trends.

2. **Scalability:**
    - The relational schema and ER diagram ensure flexibility to accommodate future expansions, such as integrating IoT devices for smart buildings.

Impact - The upgraded system will:

- Reduce vacancies by identifying leasing trends and high-demand apartment features.
- Lower operational costs through predictive maintenance alerts and utility usage optimization.
- Improve tenant retention by resolving feedback faster and ensuring compliance with safety inspections.
- Support strategic growth by analyzing referral patterns and corporate client preferences.

By bridging operational efficiency with analytical depth, this database redesign positions the company to thrive in a competitive real estate market while maintaining scalability for future innovations.