

A photograph of a modern, multi-story building with a light-colored, possibly concrete or stone, facade. The building features large windows and balconies, some with glass railings. The architecture is contemporary and geometric. The image is partially framed by a white border on the left and bottom, and a tan-colored background on the right.

FIND YOUR  
**PREMIUM  
REAL ESTATE**

**GOLDEN GATE COMPANY**



Egypt Russian University

Faculty of Management, Economic and Business Technology



*Database Systems1 project*

*IST205*

**Project Title**  
**(Real estate Company Management System)**

**Project No**  
**(2)**  
**Team Nickname**  
**(كليه under control)**

❖ **Team Members**

NO	ID	Name	Notes (By doctor)
1	224015	Mariam Tamer Abdullatif	
2	224013	Ahmed Mohmed Refaat	
3	224044	Mahmoud Mohamed Abdullatif	
4	224052	Toka Khaled Mohamed	
5	224177	Amr Mohamed Ahmed	

❖ **Notes by Doctor**

*Supervised by:*

Assoc prof. Mohamed Abd Elsalam

AL. Reham Abdallah

(Fall 2023)

## ✓ Business Rules

Golden Gate is a real estate company Whose Business is responsible for starting projects to construct buildings and sell units only. Each unit can be reserved or sold or available. Each customer can buy one unit or more, one or more unit can be bought by one customer .one or more customers can make more than one complaint, complaint must be made by at least one customer. There is an employee responsible for recording complaints. One or more customers must deal with many employees, at least one employee must deal with customer. customer can make one payment or more, Payment must be made by one customer, there are different methods for payments include (check \_ fiat money \_ Bank transfer), Customer can pay on installments or Cash. One payment must be included in one or more contracts. one or more contracts must include one payment. one or more contracts must be made by one employee, employee can make more than one contract, it is mandatory for contract to have one unit and one unit is concluded in one contract. Golden Gates company has many departments that work in them with different employees as (Sales, Manager, worker). Each department has many Employees, one or more employees can work in one department. Employee may have Dependent, it is must for dependent to have related employee .one or more Employee can work on more than one project, project must be done by one or more employee. Each Project has one or more buildings, one or more buildings must be concluded in a project. Each Building must have at least one unit, one or more unit must be in a building. One unit can have many pictures or none, picture must be for unit.

✓ **Functional Requirements:**

Users (Customers, Employees) should be able to register and create accounts.

Employees should have access control (Sales, Manager, Worker) to them.

✓ **Unit Management:**

Units should be listed with details such as availability, reservation status, and prices.

Customers should be able to reserve, purchase, or inquire about available units.

✓ **Complaint Management:**

An office employee should be able to record and manage complaints.

Customers should be able to submit complaints through their accounts.

✓ **Payment Process:**

There are multiple payment methods (check, fiat money, bank transfer) for customers.

Allow installment payments or full cash payments for units.

✓ **Employee Management:**

Employees should be linked to departments and specific projects.

There is availability for the addition of dependents for employees.

✓ **Project Management:**

Employees should be assigned to projects.

Projects should have details like status, completion level, and associated departments.

Each project should contain buildings, and each building should have units.

✓ **Contract Management:**

Employees can create contracts for unit sales to customers.

and link each contract to the unit.

✓ **Image Availability:**

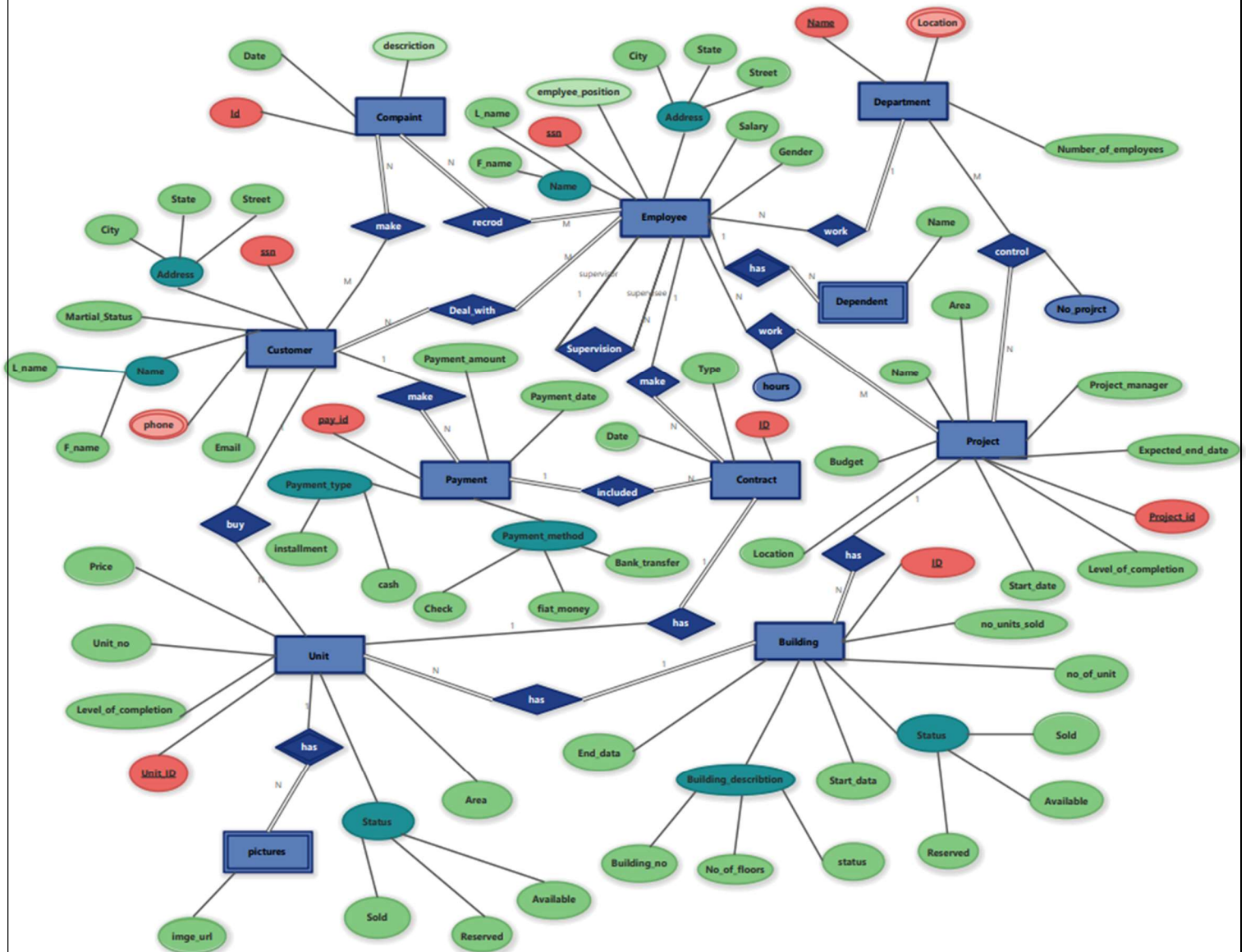
Each unit should have multiple pictures associated with it to make the customer able to see it.

✓ ***non-functional system requirements***

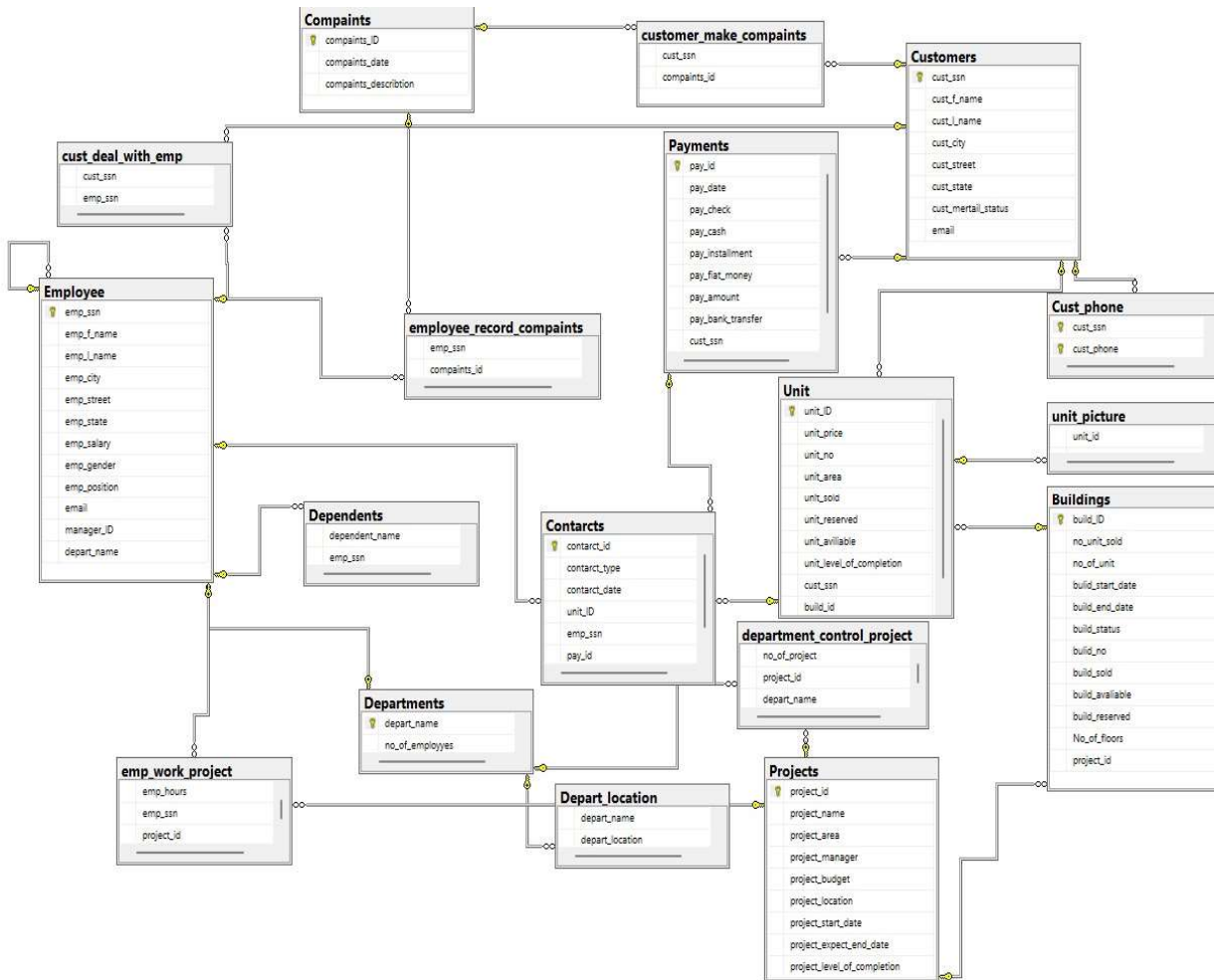
- Performance
- Security
- Scalability
- Ease of Use
- Accessibility

## ✓ Databases Design

✓ ERD



✓ Database Diagram



✓ DB schema (Mapping)

## Egypt Russian University

### Faculty of Management, Economic and Business Technology

**Customer:** (<sup>1</sup>ssnc, f\_name, l\_name, city, street, state, marital\_status, email)

**Cust\_phone:** (ssnc<sup>1</sup>, phone)

**Department:**(<sup>3</sup>department\_name, no\_of\_employees)

**Department\_location:**(department\_name<sup>3</sup>, location)

**Employee:**(<sup>2</sup>ssne, f\_name, l\_name,city, street, state, salary, gender, manger\_id<sup>2</sup>,  
deparment\_name<sup>3</sup>)

**Dependent:**(name,ssne<sup>2</sup>)

**Project:**(<sup>4</sup>project\_id, name, area, project\_manger, budget, location, start\_date,  
expect\_end\_date, level\_of\_completion)

**Building:**(<sup>5</sup>building\_id, no\_units\_sold, no\_of\_unit, sold, available, reserved, start\_date,  
end\_date, building\_no, no\_of\_floors, project\_id<sup>4</sup>)

**Unit:**(<sup>6</sup>unit\_id, price, level\_of\_completion, unit\_no, area, sold, reserved, available, ssnc<sup>1</sup>,  
building\_id<sup>5</sup>)

**Picture:** (unit\_id<sup>6</sup>,image\_url)

**Payment:**(<sup>7</sup>pay\_id, installment, fiat\_money, payment\_date, check, cash, bank\_transfer,  
payment\_amount, ssnc<sup>1</sup>,unit\_id<sup>6</sup>)

**Contract:**(<sup>8</sup>contract\_id, type, date, ssne<sup>2</sup>, pay\_id<sup>7</sup>)

**Complaints:**(<sup>8</sup>complaints\_id, date, description)

**Cust\_complaint:**(ssnc<sup>1</sup>, complaint\_id<sup>9</sup>)

**Cust\_emp:**(ssnc<sup>1</sup>, ssne<sup>2</sup>)

**Emp\_Complaints:** (ssne<sup>2</sup>, complaint\_id<sup>9</sup>)

**Emp\_proj:**(ssne<sup>2</sup>, proj\_id<sup>4</sup>, hours)

**Department\_project:**(proj\_id<sup>4</sup>, department\_name<sup>3</sup>, no\_project)

✓ **Relation**



- Many-to-Many relationship with complaint entity (one customer can make many complaint)
- Many-to-Many relationship with employee entity (one customer must deal-with one or more employee)
- many-to-one relationship with unit entity (one customer can buy one or more unit)
- many-to-one relationship with payment entity (one customer can make one or more payments)
- Many-to-Many relationship with customer entity (complaint must make by one or more customer)
- Many-to-Many relationship with employee entity (complaint must record by one or more employee)
- Many-to-Many relationship with customer entity (one employee must deal-with one or more customers)
- many-to-one relationship with supervision (one employee can manage by one manager)
- one-to-many relationship with department entity (one employee may work in department)
- many-to-one relationship with project entity (one employee may work in one or more project)
- many-to-one relationship with dependent entity (one employee may has in one or more dependent)
- many-to-one relationship with employee entity (one dependents must link to one or more employee)
- many-to-one relationship with project (one department can control one or more project)
- many-to-many relationship with department entity (one project must control by one or more department)
- many-to-one relationship with building entity (one project may has one or more building)
- one-to-many relationship with project entity (one building must has one project)

- many-to-one relationship with unit entity (one building must has one or more unit)
- one-to-many relationship with building entity (one unit must has one building)
- many-to-one relationship with picture entity (one unit may has one or pictures)
- one-to-many relationship with customer entity (one unit must sold to one customer)
- one-to-many relationship with unit entity (one picture can has one unit)
- one-to-many relationship with customer entity (one payment must make by one customer)
- many-to-many relationship with contarct entity (one payment must include one or more contarct)
- one-to-many relationship with payment entity (one contract must include one payment)
- one-to-one relationship with unit entity (one contract must has one unit)

## ✓ Database Implementation

- Alter table projects.

```
alter table Projects
alter column project_start_date date;
```

```
alter table Projects
add project_status varchar(30) not null ;
```

```
alter table Projects
drop column project_status ;
```

```
alter table Projects
add constraint unq_ex unique (project_expect_end_date);
```

```
alter table Projects
drop constraint unq_ex;
```

- Insert into Departments table.

```
insert into Departments (depart_name,
no_of_employyes) values
('HR', 20),
('Finance', 15),
('Marketing', 25),
('IT', 30),
('Sales', 18),
('Operations', 22),
('Research', 12),
('Legal', 10),
('CustomerService', 28),
('Production', 35),
('Money', 50);
```

- Insert into Unit table.

```
insert into Unit (unit_price, unit_no, unit_area, unit_sold,
unit_reserved, unit_aviliable, unit_level_of_completion, cust_ssn,
build_id)
values
(150000, 101, 1200, 'Yes', 'No', 'No', '50%', 1, 1),
(180000, 201, 1500, 'No', 'Yes', 'No', '75%', 2, 2),
(160000, 301, 1300, 'Yes', 'No', 'No', '60%', 3, 1),
(200000, 102, 1100, 'No', 'No', 'Yes', '80%', 4, 2),
(170000, 202, 1400, 'No', 'Yes', 'No', '70%', 5, 1),
(190000, 302, 1200, 'Yes', 'No', 'No', '65%', 6, 2),
```



# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
(140000, 103, 1000, 'No', 'No', 'Yes', '85%', 7, 1),  
(210000, 203, 1600, 'Yes', 'Yes', 'No', '90%', 8, 2),  
(130000, 303, 900, 'No', 'Yes', 'No', '55%', 9, 1),  
(220000, 104, 1700, 'Yes', 'No', 'No', '95%', 10, 2);
```

### ✓ Update

- Update Customers

```
UPDATE Customers  
SET cust_city =  
'NewCity'WHERE  
cust_ssn= 1;
```

- Update Cust\_phone

```
UPDATE Cust_phone  
SET cust_phone = '555-555-  
5555'WHERE cust_ssn = 1;
```

- Update Projects

```
UPDATE Projects  
SET project_budget =  
154112WHERE project_id =  
1;
```

### ✓ Delete

```
/*__Delete Data From  
Unit*/delete from Unit  
where unit_ID = 11;
```

```
/*__Delete Data From  
Buildings*/delete from Buildings  
where build_ID = 11;
```

```
/*__Delete Data From  
Payments*/delete from Payments  
where pay_id = 11;
```

```
/*__Delete Data From  
Employee*/delete from Employee  
where emp_ssn = 11;
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

### ✓ Select

```
/*__Select Data From Customer table__*/
select cust_f_name, cust_l_name, cust_city, cust_street,
       cust_state, cust_mertail_status, email
from Customers;
select distinct * from Customers;

/*__Select Data From Employee table__*/
select emp_f_name, emp_l_name, emp_city, emp_street, emp_state, emp_salary,
       email, manager_ID, depart_name
from Employee
select distinct * from Employee;

/*__Select Data From Building table__*/
select no_unit_sold, no_of_unit, bulid_start_date, build_end_date, build_status,
       bulid_no, No_of_floors, project_id
from Buildings
select distinct * from Buildings
```

### ✓ Aggregation function

```
/*__Aggregation Select Data From Unit table__*/
select count(*) as "number_of_unit" , min(unit_price) as "minimum_price" ,
       avg(unit_price) as "avg_price" , max(unit_area) as "maximum_area"
from Unit

/*__Aggregation Select Data From Employee table__*/
select count(*) as "number_of_Employee" , min(emp_salary) as "minimum_salary"
       , avg(emp_salary) as "avg_salary" , max(emp_salary) as "maximum_salary"
from Employee

/*__Aggregation Select Data From Projects table__*/
select count(*) as "number_of_Projects" , min(project_budget) as
       "minimum_project_budget" , avg(project_budget) as "avg_project_budget" ,
       max(project_area) as "maximum_project_area"
from Projects

/*__Aggregation Select Data From Buildings table__*/
select count(no_unit_sold) as "no_unit_sold" , count(no_of_unit) as "no_of_unit"
       , avg(No_of_floors) as "avg_No_of_floors"
from Buildings
```

✓ Group By and Order By and Having and select and join.

```
/*__Retrieve the department names and the count of employees in each
departmentfrom__*/
```

```
SELECT depart_name, COUNT(emp_ssn) as "number of Employee"
from Employee
group by depart_name
```

```
/*__Retrieve project managers and the count of projects they manage
__*/SELECT project_manager, COUNT(project_id) as "number of project"
from Projects
group by project_manager
having COUNT(project_id)>
1order by project_manager
```

```
/*__Retrieval for each building the number of floors it contains
__*/SELECT build_ID,No_of_floors
from Buildings
group by No_of_floors ,build_ID;
```

```
/*__Retrieve columns contract id contract date and unit ID and sort the
resultsby unit ID__*/
SELECT contract_id ,contract_date,unit_ID
from Contracts
order by unit_ID asc;
```

```
/*__Retrieve for each employee the full name along with the count of
contractsthey make, considering only
those employees with more than one contract, and sorting the results by
fullname_*/
```

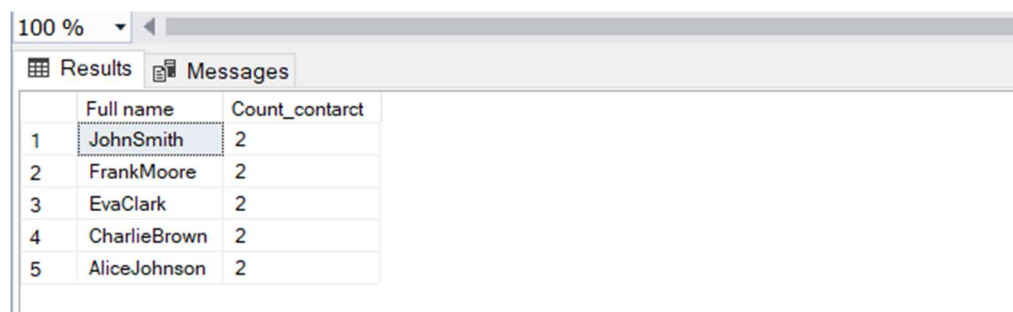
```
SELECT emp_f_name+emp_l_name as "Full name" ,COUNT(contract_id) as
"Count_contract"from Employee e join Contracts c
on e.emp_ssn =c.emp_ssn
group by emp_f_name+emp_l_name
having COUNT(contract_id)>1
order by emp_f_name+emp_l_name desc;
```

```
/*__Retrieve for each customer the full name of the maximum unit price and
totalunits and sort the data by total units descending sort__*/
SELECT cust_f_name + cust_l_name as "Full name", max(unit_price)
as"max_unit_price" ,count(unit_ID) as "count_unit"
from Customers c left join Unit
uon c.cust_ssn=u.cust_ssn
group by cust_f_name +
cust_l_nameorder by
count(unit_ID) desc;
```

Faculty of Management, Economic and Business Technology

- This SQL query joins the item, employee, contract tables together based on their foreign key relationship. It selects specific columns from the table, including the item emp\_f\_name, emp\_l\_name, contract\_id. The output will include that matching values in all two table:

```
SELECT emp_f_name+emp_l_name as "Full name", COUNT(contract_id) as "Count_contract"
from Employee e join Contracts c
on e.emp_ssn =c.emp_ssn
group by emp_f_name+emp_l_name
having COUNT(contract_id)>1
order by emp_f_name+emp_l_name desc;
```



The screenshot shows a database query result window with a zoom level of 100%. It has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'Full name' and 'Count\_contract'. The table contains five rows of data, ordered by full name in descending order. The first row is 'JohnSmith' with a count of 2, and the last row is 'AliceJohnson' with a count of 2.

	Full name	Count_contract
1	JohnSmith	2
2	FrankMoore	2
3	EvaClark	2
4	CharlieBrown	2
5	AliceJohnson	2

This query essentially lists employee full names along with the number of contracts they have, but it only displays employees who have more the one contract, ordered by full names in descending order.

Faculty of Management, Economic and Business Technology

- This query will join the "Customers" table with the "Unit" table based on their shared key (cust\_ssn). It then retrieves the full names of customers, along with the maximum unit price associated with each customer and the count of units they have. The results are grouped by full name and ordered by the count of units in descending order.

```
SELECT cust_f_name + cust_l_name as "Full name", max(unit_price) as "max_unit_price", count(unit_ID) as "count_unit"
from Customers c left join Unit u
on c.cust_ssn=u.cust_ssn
group by cust_f_name + cust_l_name
order by count(unit_ID) desc;
```

	Full name	max_unit_price	count_unit
1	AliceJohnson	160000	1
2	BobWilliams	200000	1
3	CharlieBrown	170000	1
4	DavidClark	190000	1
5	EmmaWhite	140000	1
6	FrankMiller	210000	1
7	GraceTaylor	130000	1
8	HarryWalker	220000	1
9	JaneSmith	180000	1
10	JohnDoe	150000	1

Retrieve for each customer the full name of the maximum unit price and total units and sort the data by total units descending sort.



Faculty of Management, Economic and Business Technology

- This query joins the emp\_work\_project table with the Projects table based on their shared key (project\_id). It retrieves details like the employee SSN, the count of projects they've worked on, project name, manager, location, maximum project budget, and average project area for each employee.

```
SELECT top 5 emp_ssn , count(p.project_id) ,project_name,project_manager
,project_location , max (project_budget) ,avg (project_area)
from emp_work_project w right join Projects p
on w.project_id=p.project_id
group by project_name,project_manager,project_location,emp_ssn
order by emp_ssn asc;
```

Results Messages							
	emp_ssn	(No column name)	project_name	project_manager	project_location	(No column name)	(No column name)
1	1	1	Project1	Manager1	Location1	154112	5000
2	2	1	Project2	Manager2	Location2	745725	10000
3	3	1	Project3	Manager3	Location3	7527527	18240
4	4	1	Project4	Manager1	Location4	7278524	5000
5	5	1	Project5	Manager2	Location5	752782752	5000

Retrieve information about the top 5 employees based on the count of projects they worked on, the information includes the project details such as project name, project manager project location along with the maximum project budget and average project area for each employee and sort the data by employee.

Faculty of Management, Economic and Business Technology

- This query assumes that there are foreign key relationships between the tables item, employee, and contract via the emp\_id column. Adjust the column names accordingly based on your actual table structure. This query will retrieve the specified columns and display matching values present in all three tables based on their foreign key relationships.

```
select top 5 w.emp_ssn, count(w.project_id) "Number_of_Projects_Worked",  
max(p.project_budget) "Max_Project_Budget", avg(p.project_area) "Avg_Project_Area"  
from emp_work_project w left join Projects p  
on w.project_id = p.project_id  
group by w.emp_ssn  
order by "Number_of_Projects_Worked" desc;
```

Results		Messages		
	emp_ssn	Number_of_Projects_Worked	Max_Project_Budget	Avg_Project_Area
1	6	1	72527	500074
2	5	1	752782752	5000
3	4	1	7278524	5000
4	3	1	7527527	18240
5	2	1	745725	10000

This query will give you the top 5 employees along with the count of projects they've worked on, the maximum project budget they've worked with, and the average project area for each employee based on their work within the emp\_work\_project table linked to the Projects table. Adjustments can be made as needed based on your specific requirements or additional details you'd like to retrieve.

## Egypt Russian University

### Faculty of Management, Economic and Business Technology

- This query assumes that there are foreign key relationships between the tables Payments and Customers via the cust\_ssn column. Adjust the column names accordingly based on your actual table structure. This query will retrieve the specified columns and display matching values present in both tables based on their foreign key relationships.

```
select p.pay_id, p.pay_date, p.pay_amount, c.cust_f_name, c.cust_l_name, c.email
from Payments p join Customers c
on p.cust_ssn = c.cust_ssn;
```

Results		Messages				
	pay_id	pay_date	pay_amount	cust_f_name	cust_l_name	email
1	1	2023-01-15	5000	John	Doe	john.doe@email.com
2	2	2023-02-15	6000	Jane	Smith	jane.smith@email.com
3	3	2023-03-15	7000	Alice	Johnson	alice.johnson@email.com
4	4	2023-04-15	5500	Bob	Williams	bob.williams@email.com
5	5	2023-05-15	6500	Charlie	Brown	charlie.brown@email.com
6	6	2023-06-15	7500	David	Clark	david.clark@email.com
7	7	2023-07-15	8000	Emma	White	emma.white@email.com
8	8	2023-08-15	9000	Frank	Miller	frank.miller@email.com
9	9	2023-09-15	8500	Grace	Taylor	grace.taylor@email.com
10	10	2023-10-15	9500	Harry	Walker	harry.walker@email.com

This query retrieves payment details (payment ID, date, amount) along with customer information (first name, last name, email) by joining the Payments table with the Customers table based on the customer's social security number (cust\_ssn).

✓ *code*

```
create database real_state_G2

/*_____Create Table Customers
_____*/create table Customers(
cust_ssn int primary key identity(1,1)
,cust_f_name varchar(30) not null,
cust_l_name varchar(30) not null,
cust_city varchar(30) ,
cust_street varchar(30) ,
cust_state varchar(30) ,
cust_mertail_status varchar(30) ,
email varchar(30) unique
)

/*_____Create Table Cust_phone
_____*/create table Cust_phone(
cust_ssn int,
cust_phone varchar(30) unique,
primary key (cust_ssn ,cust_phone
),
constraint cust_phone_fk foreign key
(cust_ssn)references customers(cust_ssn)
)

/*_____Create Table Projects
_____*/create table Projects(
project_id int primary key identity(1,1)
,project_name varchar(30) not null,
project_area int not null,
project_manager varchar(30) not
null,project_budget int,
project_location varchar(30) not null,
project_start_date varchar(30) not null,
project_expect_end_date varchar(30) not null,
project_level_of_completion varchar(30)
)

/*_____Create Table Departments
_____*/create table Departments(
depart_name varchar(30) primary
key,no_of_employyes int not null,
)

create table Depart_location(
depart_name varchar(30),
```

```
depart_location varchar(30),
constraint depart_location foreign key (depart_name)
references Departments(depart_name)
)

/*_____Create Table Employee
_____*/create table Employee(
emp_ssn int primary key identity(1,1)
,emp_f_name varchar(30) not null,
emp_l_name varchar(30) not null,
emp_city varchar(30) ,
emp_street varchar(30)
,emp_state varchar(30)
, emp_salary int ,
emp_gender varchar(30),
emp_position
varchar(30),email
varchar(30) unique,
manager_ID int ,
depart_name varchar(30),
foreign key (manager_ID) references
Employee(emp_ssn),constraint emp_depart_fk foreign
key (depart_name) references
Departments(depart_name)
)
```

```
/*_____Create Table Buildings
_____*/create table Buildings (
build_ID int primary key identity(1,1)
,no_unit_sold int ,
no_of_unit int,
bulid_start_date varchar(30),
build_end_date varchar(30),
build_status varchar(30),
bulid_no int not null,
build_sold varchar(30),
build_avaliable varchar(30),
build_reserved varchar(30),
No_of_floors int,
project_id int ,
constraint project_build_fk foreign key (project_id)
references Projects (project_id)
)
```

```
/*_____Create Table Payments
_____*/create table Payments (
pay_id int primary key identity(1,1)
,pay_date varchar(30) not null,
```

```
pay_check varchar (30) ,
pay_cash varchar (30),
pay_installment varchar (30),
pay_fiat_money varchar (30),
pay_amount int not null,
pay_bank_transfer varchar(30),
cust_ssn int references customers(cust_ssn)
)

/*_____Create Table Unit
_____*/create table Unit (
unit_ID int primary key identity(1,1)
,unit_price int not null,
unit_no int not null ,
unit_area int null,
unit_sold varchar(30),
unit_reserved varchar
(30),unit_aviliable
varchar(30),
unit_level_of_completion varchar(30) ,
cust_ssn int foreign key references customers
(cust_ssn),build_id int foreign key references
Buildings(build_ID)
)

/*_____Create Table Contarcts
_____*/create table Contarcts (
contarct_id int primary key identity(1,1)
,contarct_type varchar (30) not null,
contarct_date varchar(30) not null,
unit_ID int,
emp_ssn int references Employee
(emp_ssn),pay_id int references
payments(pay_id),
constraint Unit_contarct_fk foreign key (unit_ID
)references Unit (unit_ID )
)

/*_____Create Table Complaints
_____*/create table Complaints(
complaints_ID int primary key identity(1,1)
,complaints_date varchar (30) not null,
complaints_describtion varchar (30) ,

)

/*_____Create Table Dependents
_____*/create table Dependents(
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
dependent_name varchar(30) ,
emp_ssn int references Employee (emp_ssn)
)
/*____Create Table unit_picture
____*/create table unit_picture(
unit_id int foreign key references Unit
(unit_ID),unit_image image
)

/*____Create Table cust_deal_with_emp____*/
create table cust_deal_with_emp(
cust_ssn int references customers(cust_ssn),
emp_ssn int references Employee (emp_ssn)
)

/*____Create Table emp_work_project
____*/create table emp_work_project (
emp_hours int ,
emp_ssn int references Employee (emp_ssn),
project_id int references Projects
(project_id)
)

/*____Create Table department_control_project
____*/create table department_control_project(
no_of_project int ,
project_id int references Projects (project_id),
depart_name varchar(30) references Departments(depart_name),
)

/*____Create Table customer_make_complaints
____*/create table customer_make_complaints(
cust_ssn int references customers(cust_ssn),
complaints_id int references complaints(complaints_ID)
)
/*____Create Table employee_record_complaints____*/

create table employee_record_complaints(
emp_ssn int references
employee(emp_ssn),
complaints_id int references complaints(complaints_ID)
)

/*__Alter Table Projects
__*/alter table Projects
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
alter column project_start_date date;
```

```
alter table Projects
add project_status varchar(30) not null ;
```

```
alter table Projects
drop column project_status ;
```

```
alter table Projects
add constraint unq_ex unique (project_expect_end_date);
```

```
alter table Projects
drop constraint unq_ex
;
```

```
/*__Alter Table Contarcts __*/
```

```
alter table Contarcts
alter column contarct_date date not null;
```

```
/*__Alter Table Complaints
```

```
__*/alter table Complaints
alter column complaints_date date not null;
```

```
/*__Alter Table Payments
```

```
__*/alter table Payments
alter column pay_date date not null;
```

```
/*__Alter Table Units __*/
```

```
alter table Unit
alter column unit_area varchar(30)
```

```
/*_Insert into Customers table_*/
insert into Customers (cust_f_name, cust_l_name, cust_city,
    cust_street,cust_state, cust_mertail_status, email)
values
('John', 'Doe', 'City1', 'Street1', 'State1', 'Single', 'john.doe@email.com'),
('Jane', 'Smith', 'City2', 'Street2', 'State2', 'Married', 'jane.smith@email.com'),
('Alice', 'Johnson', 'City3', 'Street3', 'State3',
    'Married', 'alice.johnson@email.com'),
('Bob', 'Williams', 'City4', 'Street7', 'State4',
    'Single', 'bob.williams@email.com'),
('Charlie', 'Brown', 'City5', 'Street5', 'State5',
```



# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
'Single', 'charlie.brown@email.com'),
('David', 'Clark', 'City6', 'Street6', 'State6',
'Married', 'david.clark@email.com'),
('Emma', 'White', 'City7', 'Street7', 'State7', 'Single', 'emma.white@email.com'),
('Frank', 'Miller', 'City8', 'Street3', 'State8',
'Married', 'frank.miller@email.com'),
('Grace', 'Taylor', 'City9', 'Street2', 'State9',
'Married', 'grace.taylor@email.com'),
('Harry', 'Walker', 'City10', 'Street6', 'State10',
'Single', 'harry.walker@email.com');

/*_Insert into Cust_phone table_*/
insert into Cust_phone (cust_ssn, cust_phone)
values
(1, '123-456-7890'),
(2, '987-654-3210'),
(3, '111-222-3333'),
(4, '444-555-6666'),
(5, '777-888-9999'),
(6, '333-111-4444'),
(7, '555-666-7777'),
(8, '999-888-7777'),
(9, '111-222-5555'),
(10, '666-777-9999');

/*_Insert into Projects table_*/
insert into Projects (project_name, project_area, project_manager,
project_budget, project_location, project_start_date, project_expect_end_date)
values
('Project1', 5000, 'Manager1', 14257275, 'Location1', '2023-01-01', '2023-12-31'),
('Project2', 10000, 'Manager2', 745725, 'Location2', '2023-02-01', '2023-11-30'),
('Project3', 18240, 'Manager3', 7527527, 'Location3', '2023-03-01', '2023-11-30'),
('Project4', 5000, 'Manager1', 7278524, 'Location4', '2023-04-01', '2023-10-31'),
('Project5', 5000, 'Manager2', 752782752, 'Location5', '2023-05-01', '2023-10-30'),
('Project6', 500074, 'Manager6', 72527, 'Location6', '2023-06-01', '2023-10-29'),
('Project7', 50004, 'Manager7', 10004452, 'Location7', '2023-07-01', '2023-10-28'),
('Project8', 5074, 'Manager3', 5000000, 'Location8', '2023-08-01', '2023-10-27'),
('Project9', 5000, 'Manager9', 4785210, 'Location9', '2023-09-01', '2023-10-26'),
('Project10', 5000, 'Manager2', 1000224, 'Location10', '2023-10-01', '2023-10-25');

/*_Insert into Departments table_*/
insert into Departments (depart_name,
no_of_employyes)values
('HR', 20),
('Finance', 15),
('Marketing', 25),
('IT', 30),
('Sales', 18),
('Operations', 22),
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
( 'Research', 12),
( 'Legal', 10),
( 'CustomerService', 28),
( 'Production', 35),
( 'Money', 50);

/*_Insert into Depart_location table_ **** */
insert into Depart_location (depart_name, depart_location)
values
( 'HR', 'Location1'),
( 'Finance', 'Location2'),
( 'Marketing', 'Location3'),
( 'IT', 'Location4'),
( 'Sales', 'Location5'),
( 'Operations', 'Location6'),
( 'Research', 'Location7'),
( 'Legal', 'Location8'),
( 'CustomerService', 'Location9'),
( 'Production', 'Location10');

/*_Insert into Employee table_*/
insert into Employee (emp_f_name, emp_l_name, emp_city, emp_street,
emp_state,emp_salary, email, manager_ID, depart_name)
values
( 'John', 'Smith', 'City3', 'Street1', 'State1', 50000,
'john.smith@email.com',NULL, 'HR'),
( 'Jane', 'Doe', 'City1', 'Street6', 'State2', 60000, 'jane.doe@email.com',
1, 'HR'),
( 'Alice', 'Johnson', 'City3', 'Street3', 'State3',
55000, 'alice.johnson@email.com', 1, 'Marketing'),
( 'Bob', 'Williams', 'City3', 'Street4', 'State4', 70000,
'bob.williams@email.com',3, 'IT'),
( 'Charlie', 'Brown', 'City5', 'Street5', 'State5',
65000, 'charlie.brown@email.com', 4, 'Sales'),
( 'David', 'Miller', 'City4', 'Street2', 'State6', 80000,
'david.miller@email.com',2, 'Operations'),
( 'Eva', 'Clark', 'City5', 'Street3', 'State7', 75000, 'eva.clark@email.com',
5, 'Marketing'),
( 'Frank', 'Moore', 'City8', 'Street8', 'State8', 90000, 'frank.moore@email.com',
2, 'Legal'),
( 'Grace', 'Anderson', 'City2', 'Street1', 'State9',
85000, 'grace.anderson@email.com', 4, 'Sales'),
( 'Henry', 'Taylor', 'City1', 'Street10', 'State10',
95000, 'henry.taylor@email.com', 3, 'IT'),
( 'John', 'Smith', 'City1', 'Street1', 'State1', '50000',
'johnn.smith@email.com',NULL, null);

/*_Insert into payment table_*/
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
insert into Payments (pay_date, pay_check, pay_cash,
    pay_installment, pay_fiat_money, pay_amount, pay_bank_transfer,
    cust_ssn)
values
('2023-01-15', 'Check1', NULL, NULL, NULL, 5000, NULL, 1),
('2023-02-15', NULL, 'Cash1', NULL, NULL, 6000, NULL, 2),
('2023-03-15', NULL, NULL, 'Installment1', NULL, 7000, NULL, 3),
('2023-04-15', 'Check2', NULL, NULL, NULL, 5500, NULL, 4),
('2023-05-15', NULL, 'Cash2', NULL, NULL, 6500, NULL, 5),
('2023-06-15', NULL, NULL, 'Installment2', NULL, 7500, NULL, 6),
('2023-07-15', 'Check3', NULL, NULL, NULL, 8000, NULL, 7),
('2023-08-15', NULL, 'Cash3', NULL, NULL, 9000, NULL, 8),
('2023-09-15', NULL, NULL, 'Installment3', NULL, 8500, NULL, 9),
('2023-10-15', 'Check4', NULL, NULL, NULL, 9500, NULL, 10);

/*_Insert into Building table_*/

insert into Buildings (no_unit_sold, no_of_unit, bulid_start_date,
    build_end_date, build_status, bulid_no, No_of_floors, project_id ,
    build_reserved, build_sold, build_avaliable)
values
(5, 20, '2023-03-01', '2023-12-31', 'Under Construction', 1, 5, 1
    , '10', '50', 'yes'),
(10, 15, '2023-04-01', '2023-11-30', 'Completed', 2, 8, 2, '10', '55', 'yes'),
(7, 25, '2023-05-01', '2023-10-31', 'Under Construction', 3, 6,
    1, '50', '58', 'yes'),
(8, 18, '2023-06-01', '2023-09-30', 'In Progress', 4, 7, 2, '17', '25', 'yes'),
(6, 22, '2023-07-01', '2023-08-31', 'Completed', 5, 4, 1, '10', '50', 'yes'),
(9, 17, '2023-08-01', '2023-07-31', 'Under Construction', 6 , 6,
    2, '10', '50', 'yes'),
(4, 30, '2023-09-01', '2023-06-30', 'Completed', 7 , 3 , 1, '10', '50', 'yes'),
(11, 14, '2023-10-01', '2023-05-31', 'In Progress', 8, 5, 2, '10', '50', 'yes'),
(3, 28, '2023-11-01', '2023-04-30', 'Completed', 9, 4, 1, '10', '50', 'yes'),
(12, 12, '2023-12-01', '2023-03-31', 'Under Construction', 10, 7,
    2, '10', '50', 'yes');

/*_Insert into Unit table_*/

insert into Unit (unit_price, unit_no, unit_area, unit_sold,
    unit_reserved, unit_avaliable, unit_level_of_completion, cust_ssn,
    build_id)
values
(150000, 101, 1200, 'Yes', 'No', 'No', '50%', 1, 1),
(180000, 201, 1500, 'No', 'Yes', 'No', '75%', 2, 2),
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
(160000, 301, 1300, 'Yes', 'No', 'No', '60%', 3, 1),  
(200000, 102, 1100, 'No', 'No', 'Yes', '80%', 4, 2),  
(170000, 202, 1400, 'No', 'Yes', 'No', '70%', 5, 1),  
(190000, 302, 1200, 'Yes', 'No', 'No', '65%', 6, 2),  
(140000, 103, 1000, 'No', 'No', 'Yes', '85%', 7, 1),  
(210000, 203, 1600, 'Yes', 'Yes', 'No', '90%', 8, 2),  
(130000, 303, 900, 'No', 'Yes', 'No', '55%', 9, 1),  
(220000, 104, 1700, 'Yes', 'No', 'No', '95%', 10, 2);
```

```
/*_Insert into Contarcts table_*/
```

```
insert into Contarcts (contarct_type, contarct_date, emp_ssn, pay_id  
,unit_ID)values  
( 'Type1', '2023-01-15', 1, 1,1),  
( 'Type2', '2023-02-20', 1, 2,2),  
( 'Type3', '2023-03-25', 3, 3,1),  
( 'Type4', '2023-04-30', 5, 4,3),  
( 'Type5', '2023-05-05', 5, 5,1),  
( 'Type6', '2023-06-10', 3, 6,2),  
( 'Type7', '2023-07-15', 7, 7,5),  
( 'Type8', '2023-08-20', 8, 8,3),  
( 'Type9', '2023-09-25', 7, 9,2),  
( 'Type10', '2023-10-30', 8, 10,3);
```

```
/*_Insert into emp_work_project table_*/
```

```
insert into emp_work_project (emp_ssn,  
project_id)values  
(1, 1),  
(2, 2),  
(3, 3),  
(4, 4),  
(5, 5),  
(6, 6),  
(7, 7),  
(8, 8),  
(9, 9),  
(10, 10);
```

```
/*_____Update_____*/
```

```
/*_ Update
```

```
Customers_*/UPDATE
```

```
Customers
```

```
SET cust_city =
```

```
'NewCity'WHERE cust_ssn
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
= 1;
```

```
/*_ Update  
Cust_phone_*/UPDATE  
Cust_phone  
SET cust_phone = '555-555-  
5555'WHERE cust_ssn = 1;
```

```
/*_ Update  
Projects_*/UPDATE  
Projects  
SET project_budget =  
154112WHERE project_id =  
1;
```

```
/*_ Update  
Departments*/UPDATE  
Departments  
SET no_of_employyes = 20  
WHERE depart_name = 'Department1';  
/*_ Update  
Employee_*/UPDATE  
Employee  
SET emp_salary =  
70000WHERE emp_ssn =  
1;
```

```
/*_____Delete_____*/
```

```
/*__Delete Data From  
Unit*/delete from Unit  
where unit_ID = 11;
```

```
/*__Delete Data From  
Buildings*/delete from Buildings  
where build_ID = 11;
```

```
/*__Delete Data From  
Payments*/delete from Payments  
where pay_id = 11;
```

```
/*__Delete Data From  
Employee*/delete from Employee  
where emp_ssn = 11;
```

```
/*__Delete Data From  
Employee*/delete from Employee
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
where emp_ssn = 11;
```

```
/*__Delete Data From  
Employee*/delete from Departments  
where depart_name = 'Money';
```

```
/*_____Select_____*/
```

```
/*__Select Data From Customer table__*/  
select cust_f_name, cust_l_name, cust_city, cust_street,  
       cust_state,cust_mertail_status, email  
from Customers;  
select distinct * from Customers;
```

↗

```
/*__Select Data From Employee table__*/  
select emp_f_name, emp_l_name, emp_city, emp_street, emp_state, emp_salary,  
       email,manager_ID, depart_name  
from Employee  
select distinct * from Employee;
```

↗

```
/*__Select Data From Departments table  
____*/select depart_name, no_of_employyes  
from Departments  
select distinct * from Departments;
```

```
/*__Select Data From Building table__*/  
select no_unit_sold, no_of_unit, bulid_start_date, build_end_date, build_status,  
       bulid_no, No_of_floors, project_id  
from Buildings  
select distinct * from Buildings
```

↗

```
/*__Select Data From Unit table__*/  
  
select unit_price, unit_no, unit_area, unit_sold, unit_reserved,  
       unit_aviliable,cust_ssn, build_id  
from Unit  
select distinct * from Unit
```

↗

```
/*_____aggregation function_____*/
```

```
/*__Aggregation Select Data From Unit table__*/  
select count(*) as "number_of_unit" , min(unit_price) as "minimum_price" ,  
       avg(unit_price) as "avg_price" , max(unit_area) as "maximum_area"  
from Unit
```

↗

```
/*__Aggregation Select Data From Employee table__*/  
select count(*) as "number_of_Employee" , min(emp_salary) as "minimum_salary"
```

↗

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
, avg(emp_salary) as "avg_salary" , max(emp_salary) as "maximum_salary"
from Employee
```

```
/*__Aggregation Select Data From Projects table__*/
select count(*) as "number_of_Projects" , min(project_budget) as
    "minimum_project_budget" , avg(project_budget) as "avg_project_budget" ,
    max(project_area) as "maximum_project_area"
from Projects
```

```
/*__Aggregation Select Data From Buildings table__*/
select count(no_unit_sold) as "no_unit_sold" , count(no_of_unit) as "no_of_unit"
    , avg(No_of_floors) as "avg_No_of_floors"
from Buildings
```

```
/*_____Groub By and Order Buy and Having and selselect and join
_____*/
```

```
/*__Retrieve the department names and the count of employees in each
departmentfrom__*/
SELECT depart_name, COUNT(emp_ssn) as "number of Employee"
from Employee
group by depart_name
```

```
/*__Retrieve project managers and the count of projects they manage
____*/SELECT project_manager, COUNT(project_id) as "number of project"
from Projects
group by project_manager
having COUNT(project_id)>
1order by project_manager
/*__Retrieval for each building the number of floors it contains
____*/SELECT build_ID, No_of_floors
from Buildings
group by No_of_floors , build_ID;
```

```
/*__Retrieve columns contract id contract date and unit ID and sort the
resultsby unit ID____*/
SELECT contract_id , contract_date, unit_ID
from Contracts
order by unit_ID asc;
```

```
/*__Retrieve for each employee the full name along with the count of
contracts they make, considering only
those employees with more than one contract, and sorting the results by
```

# Egypt Russian University

## Faculty of Management, Economic and Business Technology

```
fullname_*/

SELECT emp_f_name+emp_l_name as "Full name" ,COUNT(contarct_id) as
"Count_contarct"from Employee e join Contarcts c
on e.emp_ssn =c.emp_ssn
group by emp_f_name+emp_l_name
having COUNT(contarct_id)>1
order by emp_f_name+emp_l_name desc;

/*__Retrieve for each customer the full name of the maximum unit price and
totalunits and sort the data by total units descending sort__*/
SELECT cust_f_name + cust_l_name as "Full name", max(unit_price)
as"max_unit_price" ,count(unit_ID) as "count_unit"
from Customers c left join Unit
uon c.cust_ssn=u.cust_ssn
group by cust_f_name +
cust_l_nameorder by
count(unit_ID) desc;

/*__Retrieve information about the top 5 employees based on the count of
projects they worked on, The information includes the project details such
asproject name,
project manager project location along with the maximum project budget and
average project area for each employee and sort the data by employee
_____*/
/
SELECT top 5 emp_ssn , count
(p.project_id) ,project_name,project_manager ,project_location , max
(project_budget) ,avg (project_area)
from emp_work_project w right join Projects
pon w.project_id=p.project_id
group by project_name,project_manager,project_location,emp_ssn
order by emp_ssn asc;

/*
This query retrieves payment details (payment ID, date, amount) along
withcustomer information
(first name, last name, email) by joining the Payments table with the
Customerstable
based on the customer's social security number (cust_ssn).
*/

select p.pay_id,p.pay_date, p.pay_amount, c.cust_f_name, c.cust_l_name, c.email
from Payments p join Customers c
on p.cust_ssn = c.cust_ssn;

/*_____Select Top_____*/

/*
```



Faculty of Management, Economic and Business Technology

Retrieve information about the top 5 employees based on the count of projects they worked on. ↗

The information includes the project details such as project name, project manager, project location, along with the maximum project budget and average project area for each employee. Sort the data by employee. ↗

```
*/
select top 5 w.emp_ssn, count(w.project_id) AS "Number_of_Projects_Worked", max
(p.project_budget) "Max_Project_Budget", avg(p.project_area) "Avg_Project_Area"
from emp_work_project w left join Projects
on w.project_id = p.project_id
group by w.emp_ssn
order by "Number_of_Projects_Worked" desc; ↗
```

```
/*__Retrieve the top 5 unit prices
_____*/
/select top 5 unit_price
from Unit;
```