

Hospital Management System

Prepared For: Dr. Zaki Alsubhi

Prepared By:

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Section** |
| Naif Askul | 2035052 | ZK |
| Nawaf AlGhamdi | 2037159 | ZK |
| Tareq Fuad Bajaber | 2035796 | ZK |
| Faisal Balamash | 2036827 | ZK |

April 19, 2022

**Table of Contents**

[Part 1 3](#_bookmark0)

[1) About the Domain 3](#_bookmark1)

[2) Identify the entities and relation 4](#_bookmark2)

[a) List of all the entities and their relations literally: 4](#_bookmark3)

[b) List of all the entities separately 4](#_bookmark4)

[c) Map of relations between entities 4](#_TOC_250000)

[d) List of the attributes associated with each entity 5](#_bookmark5)

[e) Identifying the candidate key, primary key, composite key, super 6](#_bookmark6)

[3) ER Diagram 7](#_bookmark7)

[4) Ten possible queries according to the ER diagram 8](#_bookmark8)

[Part 2 9](#_bookmark9)

[1)Translate ER into Schema 9](#_bookmark10)

[2)Implementation of the Schema 11](#_bookmark11)

[3)Filling data into schema 12](#_bookmark12)

[Part 3 15](#_bookmark13)

[1) Queries expressed in Algebra 15](#_bookmark14)

[2) Write those queries by using the SQL language 16](#_bookmark15)

[3) Queries that contain more complex which include the following: 16](#_bookmark16)

# Part 1

## About the Domain

A hospital is a place where injured and sick people are treated and taken care of. It ensures the delivery of maximum health services, care, cure, and preventive services for each patient. Daily hundreds of patients visit hospitals to get treated by doctors and other employees for assistance. With that amount of data circulating, we need to accumulate them in one single unit, our project provides the accumulation, organizing, and maintenance of the data for the departments of hospitals.

The domain of the database coordinates **Departments, Employees**, **Doctors**, **Patients**, and **Rooms**. We aim to improve data access, making entities receive quick answers to the data they want from the database. Data sharing because the management system creates an environment in which users have better access to more and better- managed data and users can respond quickly to changes in their environment. Data security management system reduces the data access for entities to prevent data breaches which can cause the domain time, effort, and money if the data is used wrongly. And finally, minimize data inconsistency which is when different versions of the same data appear in different places and the probability of this happening is greatly reduced in a properly designed database management system.

The hospital that uses the database management system should have a

department’s unique number, The Department must have an employee managing it, and the chosen employee can manage only one department.

The employee when assigned to the department must provide their ID, first name, last name, date of birth, salary, address, and sex. An Employee can have only one supervisor and multiple supervisees. Each employee must work in a department and a department can have multiple employees that work in it.

The doctor must have ID, first name, last name, date of birth, salary, address, and sex. Doctors must be part of a department, and the department can have multiple doctors that are part of it. Every room must have a unique numeric identifier and a location inside the hospital. Each doctor is assigned to a room and a room can be assigned to multiple doctors.

Finally, when a patient is checking in, they need to provide their ID, first name, last name, date of birth, sex, and address. Patients may attend multiple doctors, and each doctor may be attended by multiple patients during his working day.

## Identify the entities and relation

#### List of all the entities and their relations literally:

* + Each **Department** must have an ID and a name.
  + Each **Employee** should have an ID, first & last name, date of birth, salary, address, and sex.
  + Each **Doctor** should have an ID, first & last name, date of birth, salary, sex and address.
  + Each **Patient** should have an ID, first & last name, date of birth, sex, and address.
  + Each **Room** must have an ID and a location.
  + Each **Employee** must work in a **Department**.
  + Each **Department** can have multiple **Employees** that work in it.
  + One **Employee** must manage a **Department**.
  + Each **Department** must have an **Employee** managing it.
  + Each **Doctor** must be a part of a **Department**.
  + Each **Department** can have multiple **Doctors** that are part of it.
  + Each **Doctor** is assigned to a **Room**.
  + Each **Room** can be assigned to multiple **Doctors.**
  + Each **Patient** may attend multiple **Doctors**.
  + Each **Doctor** may be attended by multiple **Patients**.
  + Each **Employee** can have one **Employee** supervising him.
  + Each **Employee** can supervise multiple **Employees**.

#### List of all the entities separately

* + Department
  + Employee
  + Doctor
  + Patient
  + Room

### Map of relations between entities.

* + Employee MANAGES Department.
  + Employee WORK IN Department.
  + Employee(as Supervisor) SUPERVISOR Employee(as Supervisee)
  + Doctor PART OF Department.
  + Patient ATTEND Doctor.
  + Doctor ASSIGNED to Room.

#### List of the attributes associated with each entity

##### Department:

* + - Department ID
    - Department Name

##### Employee:

* + - Employee ID
    - Employee Name (First & Last Name)
    - Date of Birth
    - Salary
    - Address
    - Sex

##### Doctor:

* + - Doctor ID
    - Doctor Name (First & Last Name)
    - Date of Birth
    - Salary
    - Address
    - Sex

##### Patient:

* + - Patient ID
    - Patient Name (First & Last Name)
    - Date of Birth
    - Address
    - Sex

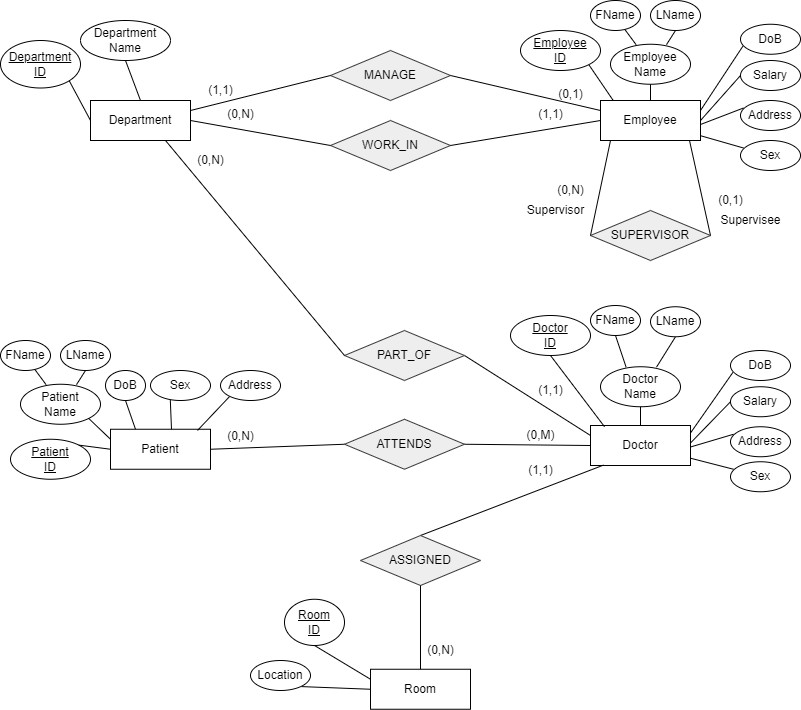
##### Room:

* + - Room ID
    - Location

#### Identifying the candidate key, primary key, composite key, super key and etc…

* + Employee ID is the primary key of Employee.
  + Department ID is a foreign key in Employee.
  + Supervisor ID is a foreign key in Employee.
  + Department ID is the primary key of Department.
  + Manager ID is a foreign Key in Department.
  + Doctor ID is the primary key of Doctor.
  + Room ID is a foreign key in Doctor.
  + Department ID is a foreign key in Doctor.
  + Patient ID is primary key in Patient.
  + Room ID is primary key of Room.
  + Patient ID & Doctor ID are composite key in Attends.

## ER Diagram



## Ten possible queries according to the ER diagram

1. **Retrieve the Doctor ID, First name, and Last Name of all doctors.**
2. **Show the records of male employees.**
3. **Show the records of employee who are managing Departments.**
4. **Show the Department ID of the Medical Department.**
5. **Create a view which retrieves the Doctor ID, first name, and last name of all doctors who are a part of Department ID 2.**
6. **Shows the Salaries of doctors that are higher than the average.**
7. **Count how many patients attend each doctor.**
8. **Show the Patients id, Doctors id, and first name who their first name matches.**
9. **Show the Room ID’s of the rooms that are not assigned to a**

**doctor.**

1. **Show Employee ID, first name, and last name of employees who are not supervising anybody.**

# Part 2

## Translate ER into Schema

###### Employee

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Employee ID** | **FName** | **LName** | **Dob** | **Salary** | **Address** | **Sex** |
| **Supervisor ID** | **Department ID** |  | | | | |

It has Employee ID as primary key and the rest of the Employee attributes. It has Department ID as foreign key due to the relation between Employee and Department is (1:N). It has Supervisor ID as foreign key due to recursive relationship and the relation is (1:N).

###### Department

|  |  |  |
| --- | --- | --- |
| **Department ID** | **Department Name** | **Manager ID** |

It has Department ID as primary key and rest of the Department attributes. Manager ID as foreign key due to the relation between the Department and Employee is (1:1).

###### Doctor

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Doctor ID** | | **FName** | **LName** | | **Dob** | **Salary** | **Address** | **Sex** |
| **Room ID** | **Department ID** | | |  | | | | |

It has Doctor ID as primary key and rest of the Doctor attributes.

It has Room ID and Department ID as foreign keys due to the relation between both of them are (1:M).

###### Patient

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Patient ID** | **FName** | **LName** | **Dob** | **Address** | **Sex** |

It has Patient ID as primary key and the rest of the Patient attributes.

###### Room

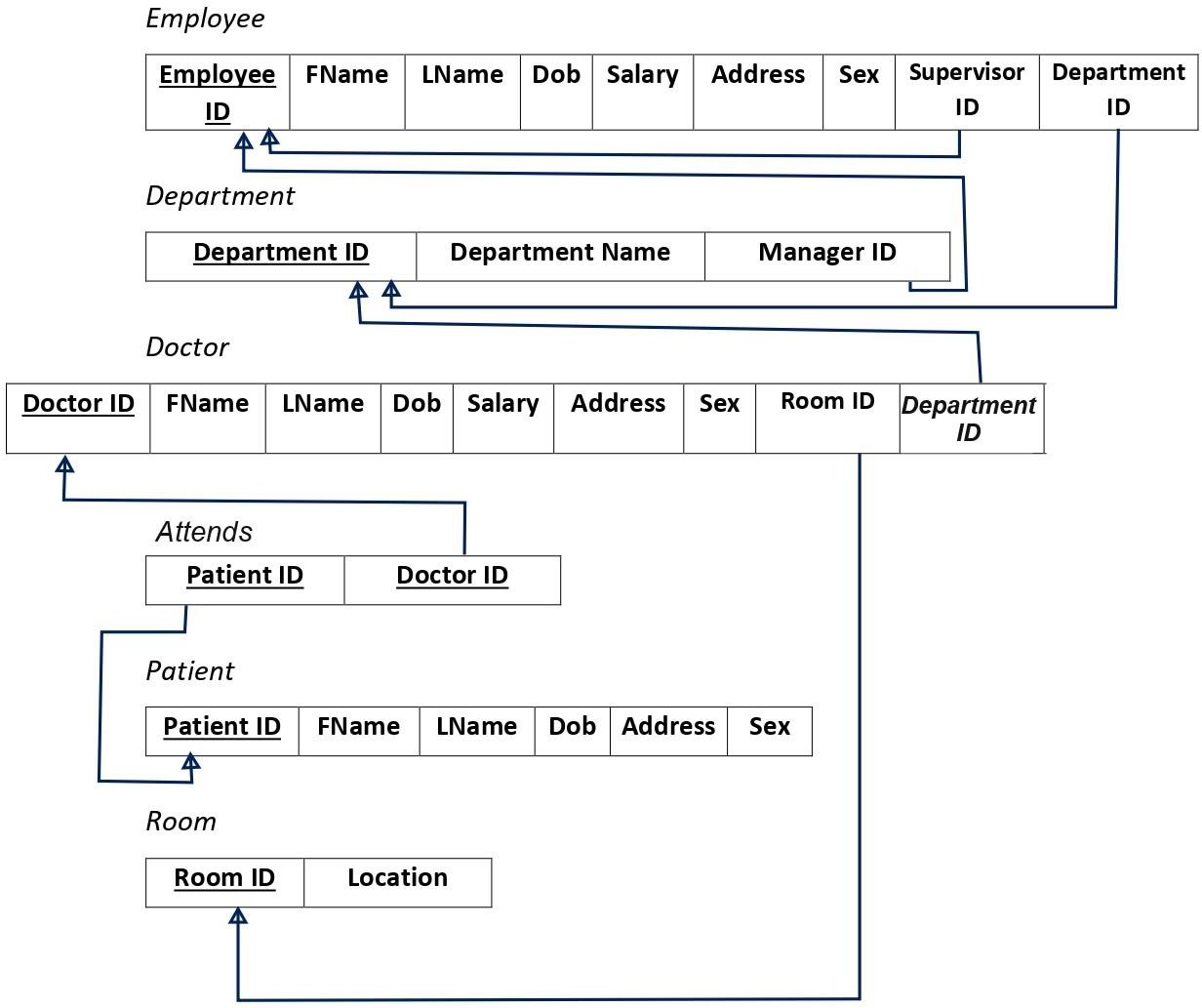
|  |  |
| --- | --- |
| **Room ID** | **Location** |

It has Room ID as primary key and Location as an attribute.

###### Attends

|  |  |
| --- | --- |
| **Patient ID** | **Doctor ID** |

It has the Patient ID and Doctor ID where they are composite key due to the relation between them (M:N).



## Implementation of the Schema

CREATE table Department(Department\_ID integer constraint D\_pk primary key, Department\_Name varchar2(25) constraint deptname unique, Manager\_ID integer);

* + Department\_ID is the primary key, and the Department\_Name must be unique.

CREATE table Employee(Employee\_ID integer constraint e\_pk primary key, Fname varchar2(15), Lname varchar2(15), Dob date, salary number(8,2), Address varchar2(25), Sex varchar2(7), Supervisor\_ID constraint svf references Employee (Employee\_ID) on delete set null, Department\_ID constraint dif references Department(Department\_ID) on delete set null);

* + Employee\_ID is the primary key, Department\_ID is a foreign key from Department,. and Supervisor\_ID is a foreign key from the sametable(Employee).

ALTER table Department add constraint dc FOREIGN KEY (Manager\_ID) references Employee(Employee\_ID) on delete set null;

* + Making the Manager\_ID foreign key from the Employee, couldn’t be done before

because the table did not exist.

CREATE table Room(Room\_ID integer constraint R\_pk primary key, Location varchar2(25));

* + Room\_ID is the primary key.

CREATE table Doctor(Doctor\_ID integer constraint Do\_pk primary key, Fname varchar2(15), Lname varchar2(15), Dob date, salary number(8,2), Address varchar2(25), Sex varchar2(7), Room\_ID constraint rif references Room(Room\_ID) on delete set null, Department\_ID constraint deptif references Department(Department\_ID) on delete set null);

* + Doctor\_ID is the primary key, Room\_ID is a foreign key from Room, Department\_ID is a foreign key from Department.

CREATE table Patient(Patient\_ID integer constraint Patient\_pk primary key, Fname varchar2(15), Lname varchar2(15), Dob date, Address varchar2(25), Sex varchar2(7));

* + Patient\_ID is the primary key.

CREATE table Attends(Patient\_ID constraint pifk references Patient(Patient\_ID) on delete set null, Doctor\_ID constraint difk references Doctor(Doctor\_ID) on delete set null, constraint Attends\_pk primary key (Patient\_ID,Doctor\_ID));

* + Patient\_ID is a foreign key from Patient, Doctor\_ID is a foreign key from Doctor, both of them are composite primary key of Attends.

## Filling data into schema

INSERT into Department values(1, 'Medical Department', null); INSERT into Department values(2, 'Medical Record Department', null); INSERT into Department values(3, 'Paramedical Department', null); INSERT into Department values(4, 'Pharmacy Department', null);

INSERT into Employee values(1, 'Naif', 'Askul', '2-APR-1990', 30000, 'Jeddah,Al Sanabel', 'Male', null, 1);

INSERT into Employee values(2,'Faisal', 'Balamash', '1-MAR-1995', 25000, 'Jeddah,Al Sharafeyah', 'Male', null, 1);

INSERT into Employee values(3,'Tareq', 'Bajaber', '15-JUN-1993', 26000, 'Jeddah,Al Safa', 'Male', null, 2);

INSERT into Employee values(4,'Nawaf', 'Alghamdi', '30-SEP-1993', 24000, 'Jeddah,Al Safa', 'Male', null, 2);

INSERT into Employee values(5,'Ahmed', 'Alghamdi', '2-APR-1990', 15000, 'Jeddah,Al Sanabel', 'Male', null, 2);

INSERT into Employee values(6,'Ahmed', 'Alzahrani', '2-APR-1991', 13000, 'Jeddah,Al naseem', 'Male', null, 3);

INSERT into Employee values(7,'Sarah', 'Alharbi', '2-APR-1991', 3000, 'Jeddah,Al naseem', 'Female', null, 4);

INSERT into Employee values(8,'Rahaf', 'Alzahrani', '17-MAY-1989', 9000, 'Jeddah,Al Sanabel', 'Female', null, 4);

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UPDATE** | **Department** | **SET** | **Manager\_ID** | **=** | **3** | **WHERE** | **department\_id** | **=** | **1;** |
| **UPDATE** | **Department** | **SET** | **Manager\_ID** | **=** | **2** | **WHERE** | **department\_id** | **=** | **2;** |
| **UPDATE** | **Department** | **SET** | **Manager\_ID** | **=** | **4** | **WHERE** | **department\_id** | **=** | **3;** |
| **UPDATE** | **Department** | **SET** | **Manager\_ID** | **=** | **1** | **WHERE** | **department\_id** | **=** | **4;** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UPDATE** | **Employee** | **SET** | **Supervisor\_ID** | **=** | **1** | **WHERE** | **Employee\_ID** | **=** | **2;** |
| **UPDATE** | **Employee** | **SET** | **Supervisor\_ID** | **=** | **1** | **WHERE** | **Employee\_ID** | **=** | **3;** |
| **UPDATE** | **Employee** | **SET** | **Supervisor\_ID** | **=** | **1** | **WHERE** | **Employee\_ID** | **=** | **4;** |
| **UPDATE** | **Employee** | **SET** | **Supervisor\_ID** | **=** | **4** | **WHERE** | **Employee\_ID** | **=** | **5;** |
| **UPDATE** | **Employee** | **SET** | **Supervisor\_ID** | **=** | **4** | **WHERE** | **Employee\_ID** | **=** | **6;** |

UPDATE Employee SET Supervisor\_ID = 2 WHERE Employee\_ID = 7; UPDATE Employee SET Supervisor\_ID = 3 WHERE Employee\_ID = 8;

INSERT into Room values(1,'Floor 2,J1'); INSERT into Room values(2,'Floor 1,E5'); INSERT into Room values(3,'Floor 1,G5'); INSERT into Room values(4,'Floor 2,GG');

INSERT into Doctor values(1, 'Zaki', 'Alsubhi', '2-APR-1990', 23000, 'Jeddah,Al Sanabel', 'Male', 1, 1);

INSERT into Doctor values(2 , 'Marwan', 'Khalil', '1-MAR- 1995', 25000, 'Jeddah,Al Sharafeyah', 'Male', 2, 2);

INSERT into Doctor values(3 , 'Radahn', 'Almalki', '15-JUN- 1993', 15000, 'Jeddah,Al Safa', 'Male', 3, 3);

INSERT into Doctor values(4 ,'Nawaf','ALzahrani ','30- SEP-1993',22000,'Jeddah,Al Safa', 'Male', 4, 4);

INSERT into patient values(1, 'Ahmed', 'AlGhamdi', '2-APR-1990', 'Jeddah,Al Sanabel', 'Male');

INSERT into patient values(2, 'Ahmed', 'bader', '9-APR-1990', 'Jeddah,Al Sanabel', 'Male');

INSERT into patient values(3, 'Bader', 'saleh', '9-SEP-1987', 'Jeddah,Al Safa', 'Male');

INSERT into patient values(4, 'Sara', 'saleh', '10-SEP-1987', 'Jeddah,Al Safa', 'Female');

INSERT into Attends values (1,1); INSERT into Attends values (2,2);

INSERT into Attends values (3,3); INSERT into Attends values (4,4);

# Part 3

## Queries expressed in algebra form

### Select Doctor\_ID, Fname, Lname from Doctor;

Π (Doctor\_ID, Fname, Lname)(Doctor)

### Select \* From Employee where sex = ‘Male’;

σ (sex = 'Male') (Employee)

### Select \* from Employee where Employee\_ID in (select

**Manager\_ID from department);**

σ (Employee\_ID IN (Π (Manager\_ID)Department))

### Select Department\_ID from department WHERE department\_name = 'Medical Department';

Π (Department\_ID) (σ (Department\_name = ‘Medical Department’)(Department))

### Create VIEW MED\_INS as Select Doctor\_ID, fname, lname from Doctor where department\_ID = 2;

Create view MED\_INS as Π (Doctor\_ID, Fname, Lname) ( σ (Department\_ID = 2)

. (Doctor))

### Select Doctor\_ID, Salary from Doctor where Salary > (Select avg(Salary) from Doctor);

σ (Salary > ( Π(ℱAVG Salary)(Doctor) ) (Π (Doctor\_ID, Salary) (Doctor) )

### Select Doctor\_ID, COUNT(Patient\_ID) from Attends Group by Doctor\_ID;

Doctor\_ID ℱ COUNT Patient\_ID (Attends)

##### Select D.Doctor\_ID, D.Fname, P.Patient\_ID, , P.Fname from Doctor D, Patient P where D.Fname = P.Fname;

Π(D.Doctor\_ID, D.FName, P.Patient\_ID, P.FName)(Doctor ⋈ Patient)

D.Fname = P.FName

* 1. **Select Room\_ID from Room minus Select Room\_ID from Doctor;**

Π (Room\_ID)(Room) − Π (Room\_ID)(Doctor)

* 1. **Select Employee\_ID, Address from Employee where supervisor\_ID is null;**

Π(Employee\_ID, Address)( σ(Supervisor\_ID = null)(Employee))

## Write those queries by using the SQL language

* 1. **Select Doctor\_ID, Fname, Lname from Doctor;**
  2. **Select \* From Employee where sex = ‘Male’;**
  3. **Select \* from Employee where Employee\_ID in (select Manager\_ID from department);**
  4. **Select Department\_ID from department WHERE department\_name = 'Medical Department';**
  5. **Create VIEW MED\_INS as Select Doctor\_ID, fname, lname from Doctor where department\_ID = 2;**
  6. **Select Doctor\_ID, Salary from Doctor where Salary > (Select avg(Salary) from Doctor);**
  7. **Select Doctor\_ID, COUNT(Patient\_ID) from Attends Group by Doctor\_ID;**
  8. **Select D.Doctor\_ID, D.Fname, P.Patient\_ID, P.Fname from Doctor D, Patient P where D.Fname = P.Fname;**
  9. **Select Room\_ID from Room minus Select Room\_ID from Doctor;**
  10. **Select Employee\_ID, Address from Employee where supervisor\_ID is null;**

## Queries that contain more complex which include the following:

1. **Selection from different tables at one time.**

SELECT E.Fname, D.Department\_name FROM Employee E, Department D WHERE E.Department\_id = D.Department\_id;

## Aggregation by using GROUP BY and HAVING statement.

SELECT Doctor\_ID, MAX(salary), AVG(salary) from doctor GROUP BY Doctor\_ID HAVING MIN(salary) < 20000;

## Nesting with aggregation

SELECT AVG(salary) FROM (SELECT salary FROM employee WHERE salary <= (SELECT AVG(salary) FROM employee));

## Nesting involving NOT IN or NOT EXISTS

SELECT Doctor\_ID, Fname, Lname, Department\_ID FROM Doctor WHERE Doctor\_ID NOT IN (Select department\_ID from department WHERE department\_name = 'Medical Department');

## Outer Join

SELECT E.Employee\_ID, E.Employee\_Name D.Department\_name, D.Department\_ID, D.Manager\_ID from Employee E, Department D where E.Employee\_ID = D.Manager\_ID(+);