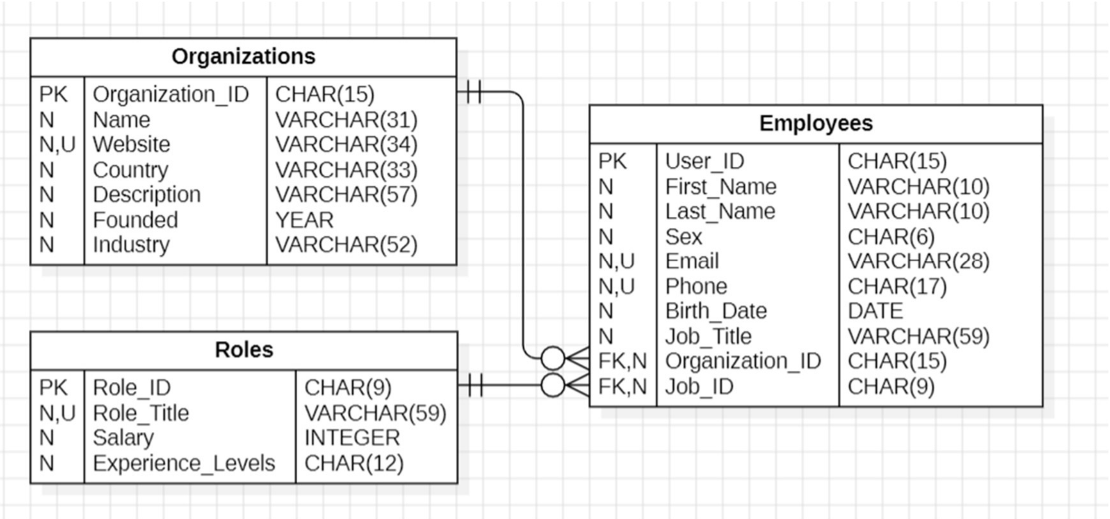
DB_ CourseWork2

Stu_ID:202118010418

1) Examine the files and determine an appropriate relationship model between them. Create and document a physical data model detailing the field types and relationships.

Answer:



Organizations

PK	Organization_ID	CHAR(15)
N	Name	VARCHAR(31)
N,U	Website	VARCHAR(34)
N	Country	VARCHAR(33)
N	Description	VARCHAR(57)
N	Founded	YEAR
N	Industry	VARCHAR(52)

Employees

PK	User_ID	CHAR(15)
N	First_Name	VARCHAR(10)
N	Last_Name	VARCHAR(10)
N	Sex	CHAR(6)
N,U	Email	VARCHAR(28)
N,U	Phone	CHAR(17)
N	Birth_Date	DATE
N	Job_Title	VARCHAR(59)
FK,N	Organization_ID	CHAR(15)
FK,N	Job_ID	CHAR(9)

Roles

PK	Role_ID	CHAR(9)
N,U	Role_Title	VARCHAR(59)
N	Salary	INTEGER
N	Experience_Levels	CHAR(12)

Note:

1. In the original CSV file, the Roles table apparently had duplicate records, so it could not be fully loaded into the Roles database table with the primary key constraint. Therefore, in the process of loading the data in the second step, I will ignore the duplicate records to load the data into the database normally, which is implemented in Question_2.

2) Load all three tables into a new database schema.

Answer:

```
create table cw_2.Roles (
Role_ID char(9) primary key,
Role_Title varchar(59) not null,
Salary integer not null,
Experience_Levels char(12) not null
);
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Roles.csv'
```

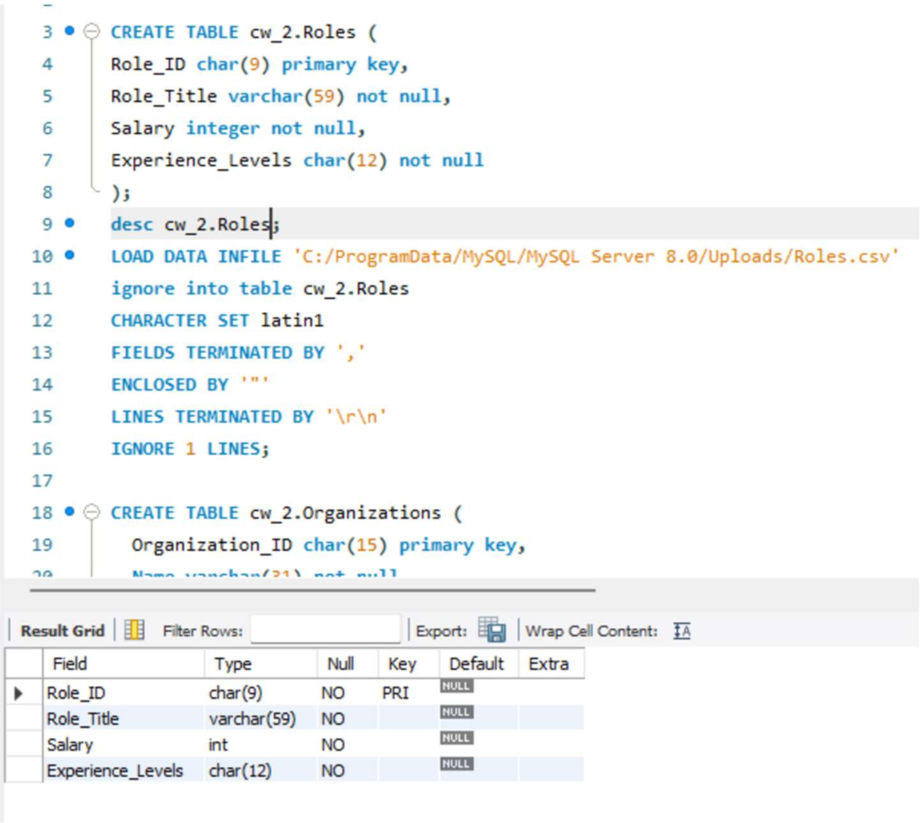
```
ignore into table cw_2.Roles
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

(Note:

1. Use User_ID as the primary key.
2. Use 'ignore into' when loading data to prevent duplicate records from entering the database table.)

```
CREATE TABLE cw_2.Organizations (
  Organization_ID char(15) primary key,
  Name varchar(31) not null,
  Website varchar(34) not null unique,
  Country varchar(33) not null,
  Description varchar(57) not null,
  Founded year not null,
  Industry varchar(52) not null
);
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Organizations.csv'
INTO TABLE cw_2.Organizations
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

CREATE TABLE cw_2.Employees (
  User_ID char(15) primary key,
  First_Name varchar(10) not null,
  Last_Name varchar(10) not null,
  Sex char(6) not null,
  Email varchar(28) not null unique,
  Phone char(17) not null unique,
  Birth_Date date not null,
  Job_Title varchar(59) not null,
  Organization_ID char(15),
  Job_ID char(9),
  foreign key (Organization_ID) references cw_2.organizations(Organization_ID),
  foreign key (Job_ID) references cw_2.Roles(Role_ID)
);
```

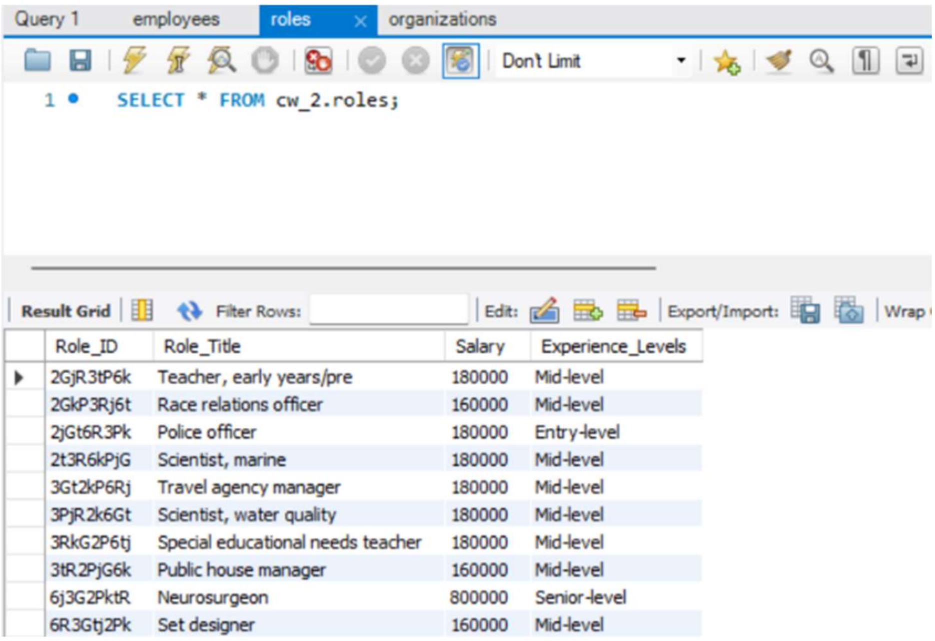
```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Employees.csv'
INTO TABLE cw_2.Employees
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(User_ID,First_Name, Last_Name, Sex, Email, Phone, @date_time_variable, Job_Title,
Organization_ID, Job_ID)
SET Birth_Date = STR_TO_DATE(@date_time_variable, '%d/%m/%Y');
```



```
CREATE TABLE cw_2.Roles (
Role_ID char(9) primary key,
Role_Title varchar(59) not null,
Salary integer not null,
Experience_Levels char(12) not null
);
desc cw_2.Roles;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Roles.csv'
ignore into table cw_2.Roles
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

CREATE TABLE cw_2.Organizations (
  Organization_ID char(15) primary key,
  Name varchar(31) not null
```

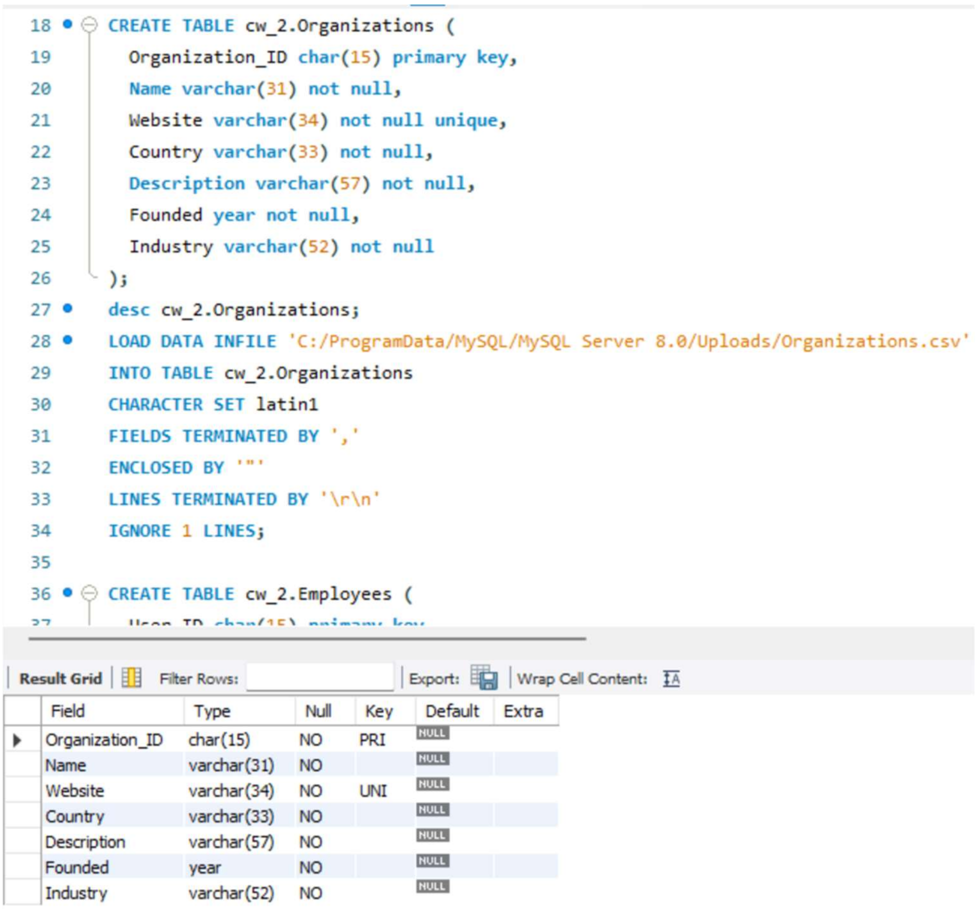
Field	Type	Null	Key	Default	Extra
Role_ID	char(9)	NO	PRI	NULL	
Role_Title	varchar(59)	NO		NULL	
Salary	int	NO		NULL	
Experience_Levels	char(12)	NO		NULL	

202118010418



```
SELECT * FROM cw_2.roles;
```

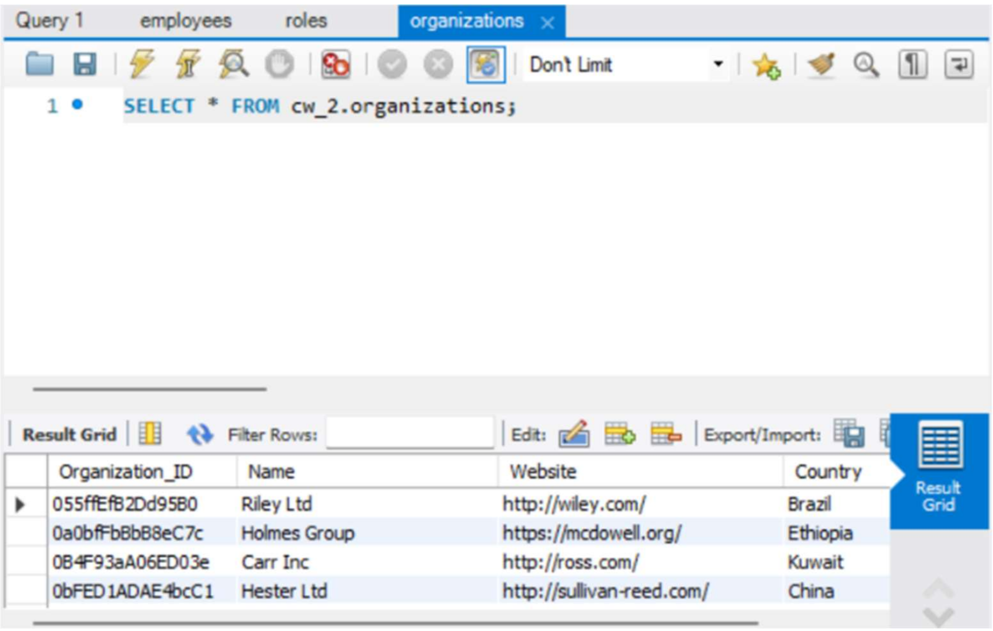
Role_ID	Role_Title	Salary	Experience_Levels
2GjR3tP6k	Teacher, early years/pre	180000	Mid-level
2GkP3Rj6t	Race relations officer	160000	Mid-level
2jGt6R3Pk	Police officer	180000	Entry-level
2t3R6kPjG	Scientist, marine	180000	Mid-level
3Gt2kP6Rj	Travel agency manager	180000	Mid-level
3PjR2k6Gt	Scientist, water quality	180000	Mid-level
3RkG2P6tj	Special educational needs teacher	180000	Mid-level
3tR2PjG6k	Public house manager	160000	Mid-level
6j3G2PktR	Neurosurgeon	800000	Senior-level
6R3Gtj2Pk	Set designer	160000	Mid-level



```
CREATE TABLE cw_2.Organizations (
  Organization_ID char(15) primary key,
  Name varchar(31) not null,
  Website varchar(34) not null unique,
  Country varchar(33) not null,
  Description varchar(57) not null,
  Founded year not null,
  Industry varchar(52) not null
);
desc cw_2.Organizations;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Organizations.csv'
INTO TABLE cw_2.Organizations
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

CREATE TABLE cw_2.Employees (
  User_ID char(15) primary key
```

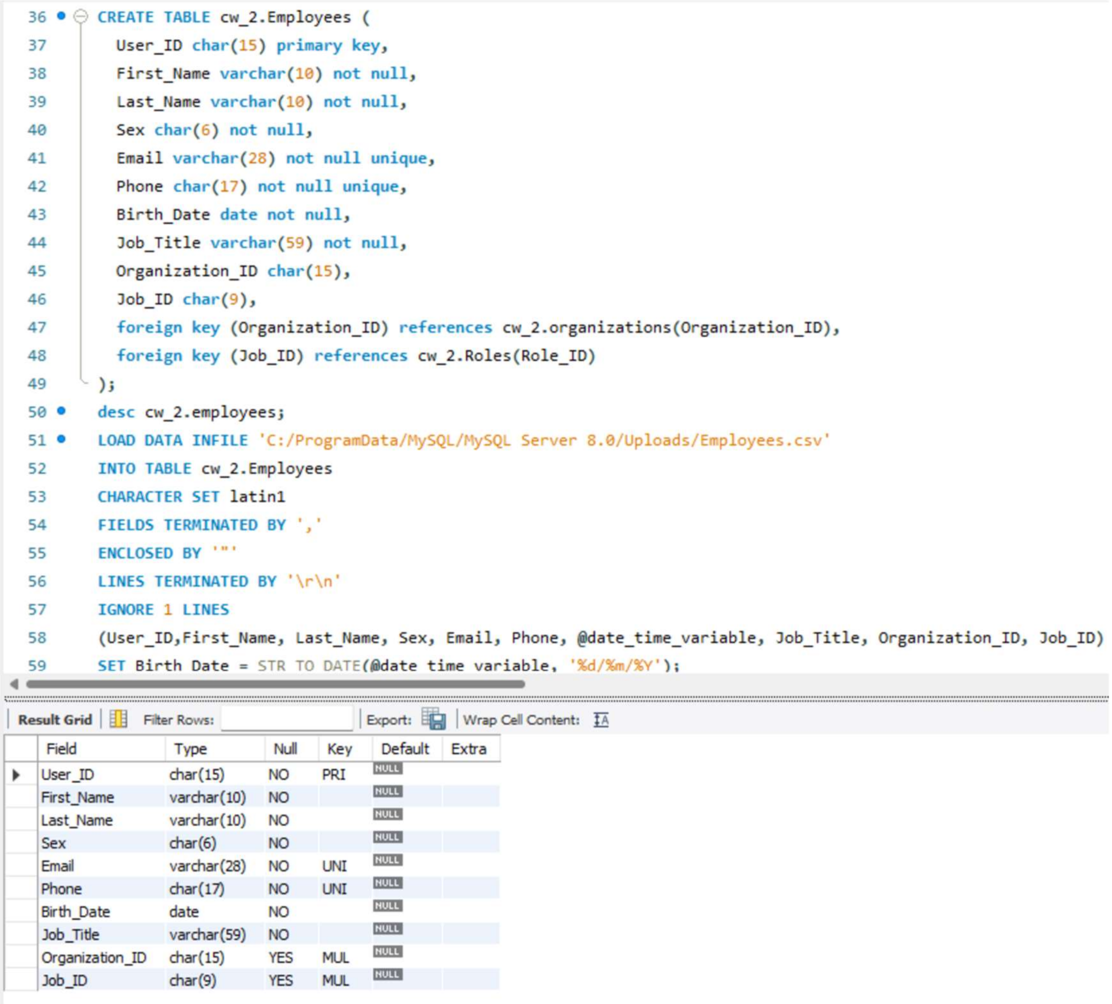
Field	Type	Null	Key	Default	Extra
Organization_ID	char(15)	NO	PRI	NULL	
Name	varchar(31)	NO		NULL	
Website	varchar(34)	NO	UNI	NULL	
Country	varchar(33)	NO		NULL	
Description	varchar(57)	NO		NULL	
Founded	year	NO		NULL	
Industry	varchar(52)	NO		NULL	

202118010418



```sql
SELECT * FROM cw_2.organizations;
```

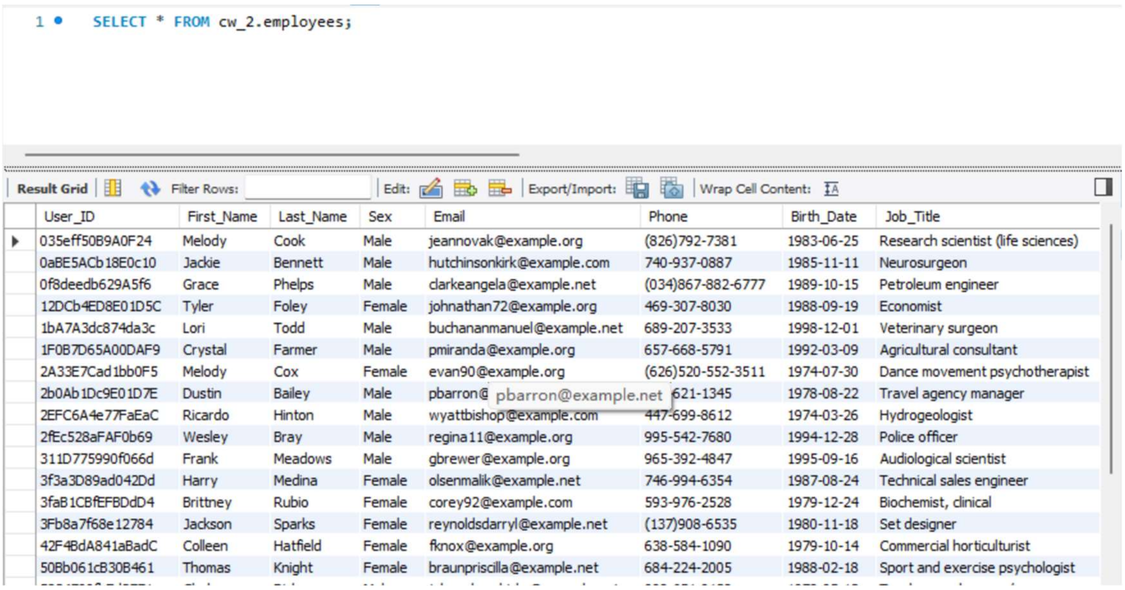
Organization_ID	Name	Website	Country
055ffEfB2Dd95B0	Riley Ltd	http://wiley.com/	Brazil
0a0bfFbBbB8eC7c	Holmes Group	https://mcdowell.org/	Ethiopia
0B4F93aA06ED03e	Carr Inc	http://ross.com/	Kuwait
0bFED1ADAE4bcC1	Hester Ltd	http://sullivan-reed.com/	China



```sql
CREATE TABLE cw_2.Employees (
  User_ID char(15) primary key,
  First_Name varchar(10) not null,
  Last_Name varchar(10) not null,
  Sex char(6) not null,
  Email varchar(28) not null unique,
  Phone char(17) not null unique,
  Birth_Date date not null,
  Job_Title varchar(59) not null,
  Organization_ID char(15),
  Job_ID char(9),
  foreign key (Organization_ID) references cw_2.organizations(Organization_ID),
  foreign key (Job_ID) references cw_2.Roles(Role_ID)
);
desc cw_2.employees;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Employees.csv'
INTO TABLE cw_2.Employees
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(User_ID,First_Name, Last_Name, Sex, Email, Phone, @date_time_variable, Job_Title, Organization_ID, Job_ID)
SET Birth_Date = STR_TO_DATE(@date_time_variable, '%d/%m/%Y');
```

Field	Type	Null	Key	Default	Extra
User_ID	char(15)	NO	PRI	NULL	
First_Name	varchar(10)	NO		NULL	
Last_Name	varchar(10)	NO		NULL	
Sex	char(6)	NO		NULL	
Email	varchar(28)	NO	UNI	NULL	
Phone	char(17)	NO	UNI	NULL	
Birth_Date	date	NO		NULL	
Job_Title	varchar(59)	NO		NULL	
Organization_ID	char(15)	YES	MUL	NULL	
Job_ID	char(9)	YES	MUL	NULL	

202118010418

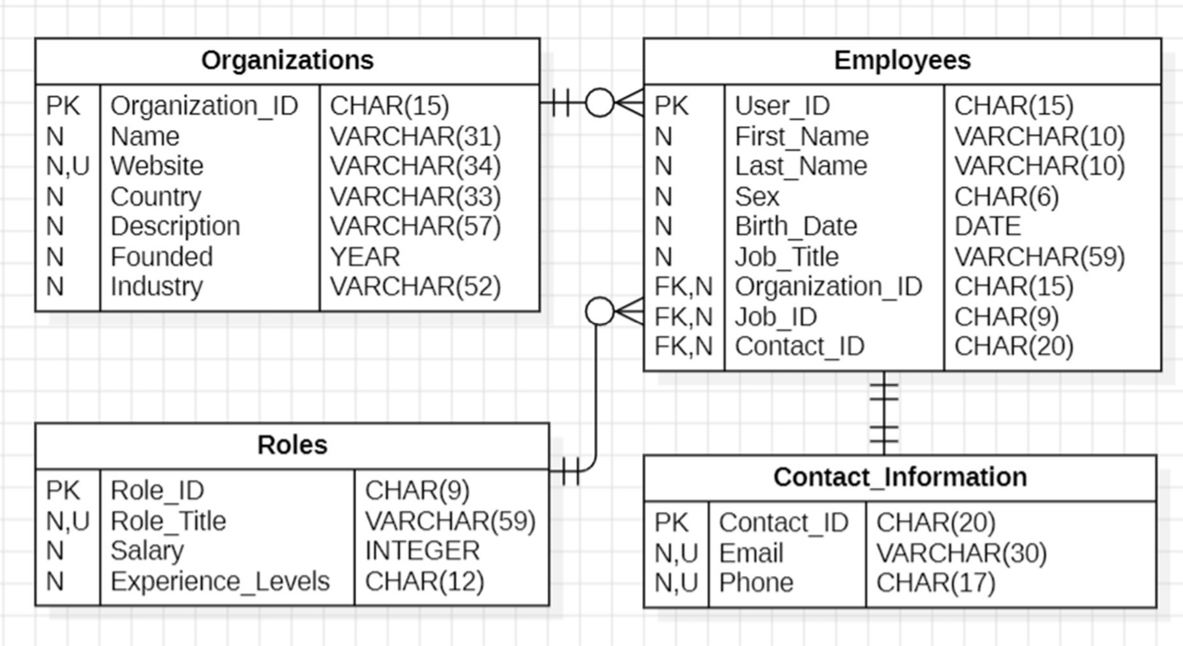


```
SELECT * FROM cw_2.employees;
```

User_ID	First_Name	Last_Name	Sex	Email	Phone	Birth_Date	Job_Title
035eff50B9A0F24	Melody	Cook	Male	jeannovak@example.org	(826)792-7381	1983-06-25	Research scientist (life sciences)
0aBE5ACb18E0c10	Jackie	Bennett	Male	hutchinsonkirk@example.com	740-937-0887	1985-11-11	Neurosurgeon
0f8deedb629A5f6	Grace	Phelps	Male	clarkeangela@example.net	(034)867-882-6777	1989-10-15	Petroleum engineer
12DCb4ED8E01D5C	Tyler	Foley	Female	johnathan72@example.org	469-307-8030	1988-09-19	Economist
1bA7A3dc874da3c	Lori	Todd	Male	buchananmanuel@example.net	689-207-3533	1998-12-01	Veterinary surgeon
1F0B7D65A00DAF9	Crystal	Farmer	Male	pmiranda@example.org	657-668-5791	1992-03-09	Agricultural consultant
2A33E7Cad1bb0F5	Melody	Cox	Female	evan90@example.org	(626)520-552-3511	1974-07-30	Dance movement psychotherapist
2b0Ab1Dc9E01D7E	Dustin	Bailey	Male	pbarron@ pbarron@example.net	621-1345	1978-08-22	Travel agency manager
2EFC6A4e77FaEaC	Ricardo	Hinton	Male	wyattbishop@example.com	447-699-8612	1974-03-26	Hydrogeologist
2fEc528aFAF0b69	Wesley	Bray	Male	regina11@example.org	995-542-7680	1994-12-28	Police officer
311D775990f066d	Frank	Meadows	Male	gbrewer@example.org	965-392-4847	1995-09-16	Audiological scientist
3f3a3D89ad042Dd	Harry	Medina	Female	olsenmalik@example.net	746-994-6354	1987-08-24	Technical sales engineer
3faB1CBfEFBDdD4	Brittney	Rubio	Female	corey92@example.com	593-976-2528	1979-12-24	Biochemist, clinical
3Fb8a7f68e12784	Jackson	Sparks	Female	reynoldsdarryl@example.net	(137)908-6535	1980-11-18	Set designer
42F4BdA841aBadC	Colleen	Hatfield	Female	fknox@example.org	638-584-1090	1979-10-14	Commercial horticulturist
50Bb061cB30B461	Thomas	Knight	Female	braunpriscilla@example.net	684-224-2005	1988-02-18	Sport and exercise psychologist

3) Recommend and design one additional table that would be appropriate for more efficient reporting of relevant data.

Answer:



Organizations

PK	Organization_ID	CHAR(15)
N	Name	VARCHAR(31)
N,U	Website	VARCHAR(34)
N	Country	VARCHAR(33)
N	Description	VARCHAR(57)
N	Founded	YEAR
N	Industry	VARCHAR(52)

Employees

PK	User_ID	CHAR(15)
N	First_Name	VARCHAR(10)
N	Last_Name	VARCHAR(10)
N	Sex	CHAR(6)
N	Birth_Date	DATE
N	Job_Title	VARCHAR(59)
FK,N	Organization_ID	CHAR(15)
FK,N	Job_ID	CHAR(9)
FK,N	Contact_ID	CHAR(20)

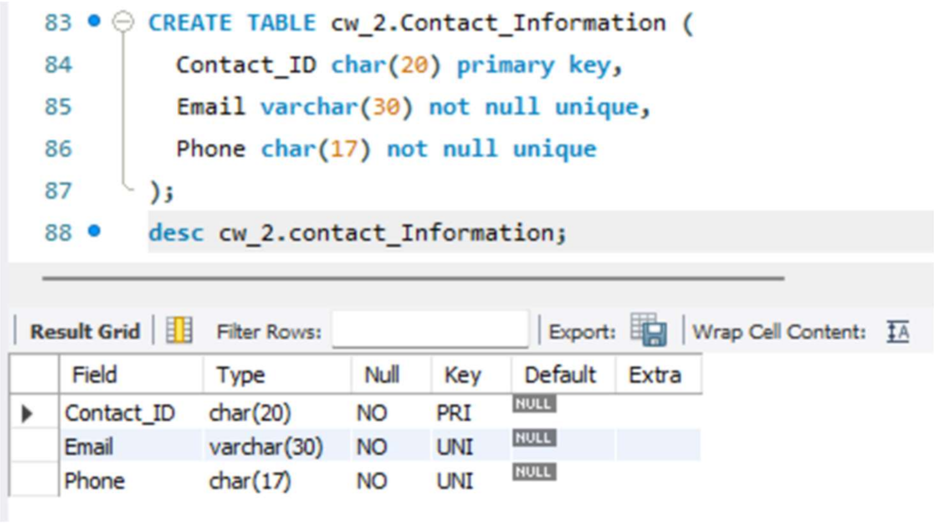
Roles

PK	Role_ID	CHAR(9)
N,U	Role_Title	VARCHAR(59)
N	Salary	INTEGER
N	Experience_Levels	CHAR(12)

Contact_Information

PK	Contact_ID	CHAR(20)
N,U	Email	VARCHAR(30)
N,U	Phone	CHAR(17)

```
CREATE TABLE cw_2.Contact_Information (
  Contact_ID char(20) primary key,
  Email varchar(30) not null unique,
  Phone char(17) not null unique
);
```

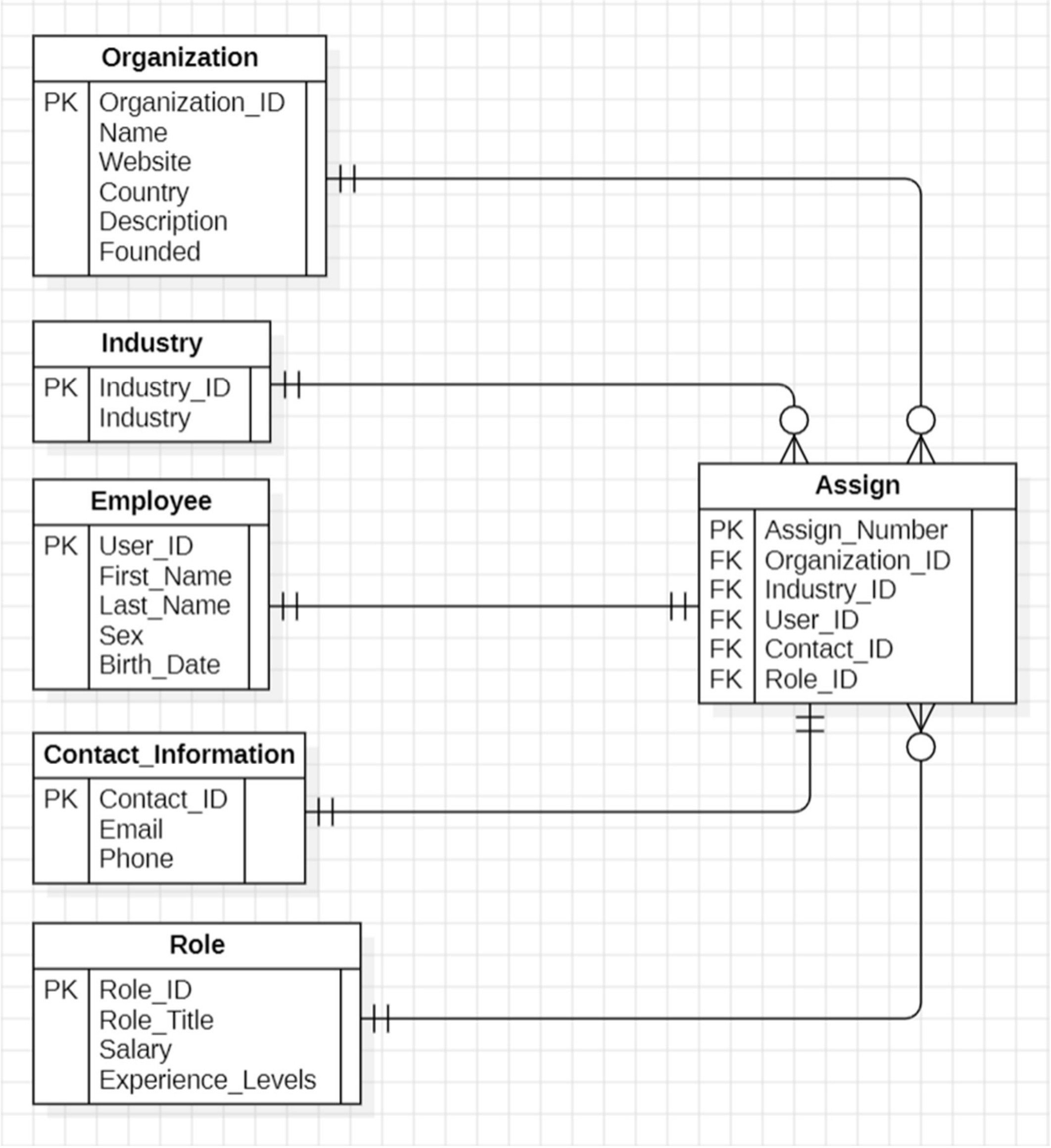


```
CREATE TABLE cw_2.Contact_Information (
  Contact_ID char(20) primary key,
  Email varchar(30) not null unique,
  Phone char(17) not null unique
);
desc cw_2.contact_Information;
```

Field	Type	Null	Key	Default	Extra
Contact_ID	char(20)	NO	PRI	NULL	
Email	varchar(30)	NO	UNI	NULL	
Phone	char(17)	NO	UNI	NULL	

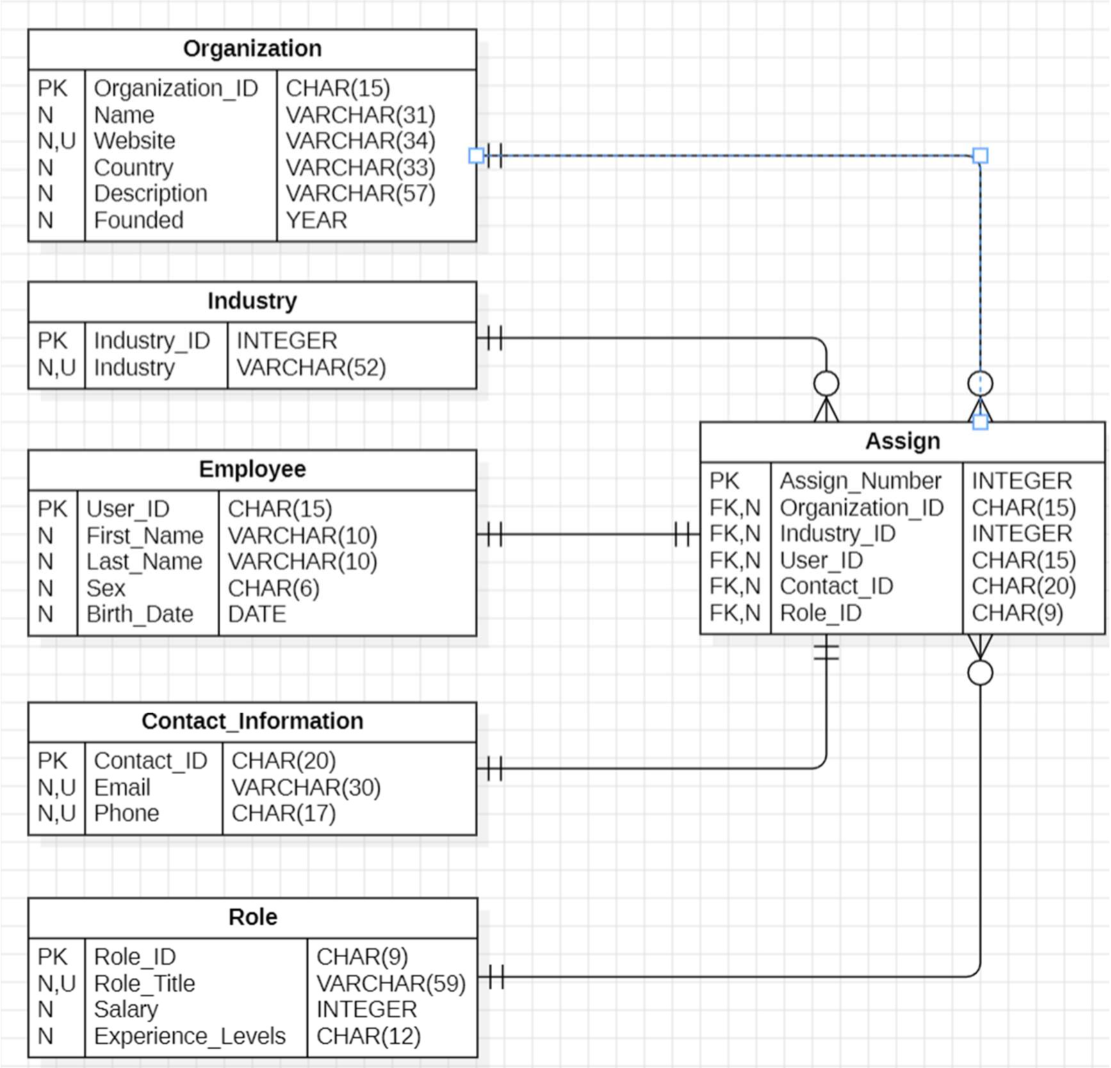
4) Redesign the data model to become more efficient for querying by detailing a logical data model showing elements of first and second normal forms where appropriate.

Answer:



5) Create and document a new physical data model detailing the new structure, and create that structure within your database schema.

Answer:



Organization

PK	Organization_ID	CHAR(15)
N	Name	VARCHAR(31)
N,U	Website	VARCHAR(34)
N	Country	VARCHAR(33)
N	Description	VARCHAR(57)
N	Founded	YEAR

Industry

PK	Industry_ID	INTEGER
N,U	Industry	VARCHAR(52)

Employee

PK	User_ID	CHAR(15)
N	First_Name	VARCHAR(10)
N	Last_Name	VARCHAR(10)
N	Sex	CHAR(6)
N	Birth_Date	DATE

Contact_Information

PK	Contact_ID	CHAR(20)
N,U	Email	VARCHAR(30)
N,U	Phone	CHAR(17)

Role

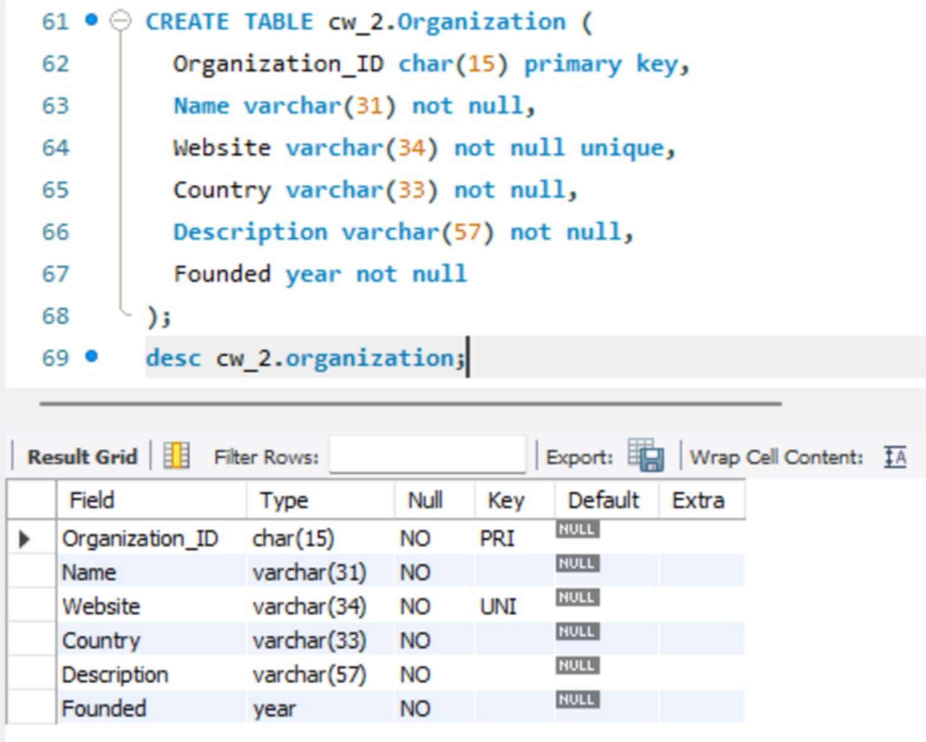
PK	Role_ID	CHAR(9)
N,U	Role_Title	VARCHAR(59)
N	Salary	INTEGER
N	Experience_Levels	CHAR(12)

Assign

PK	Assign_Number	INTEGER
FK,N	Organization_ID	CHAR(15)
FK,N	Industry_ID	INTEGER
FK,N	User_ID	CHAR(15)
FK,N	Contact_ID	CHAR(20)
FK,N	Role_ID	CHAR(9)

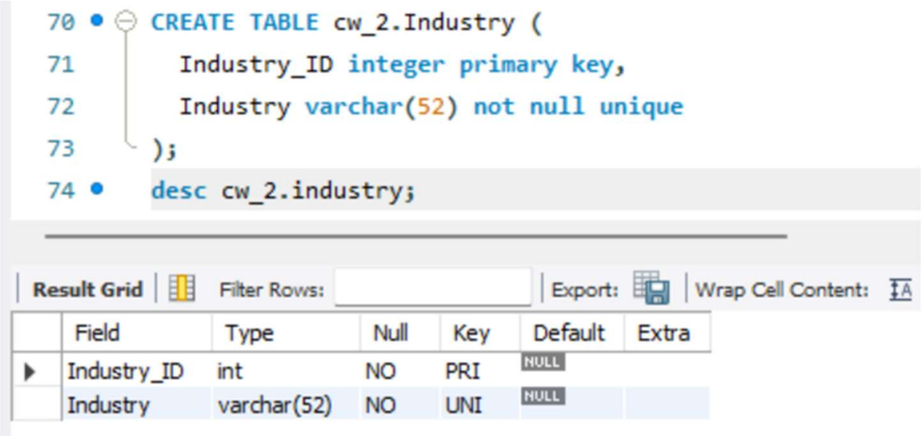
```
CREATE TABLE cw_2.Organization (
  Organization_ID char(15) primary key,
  Name varchar(31) not null,
  Website varchar(34) not null unique,
  Country varchar(33) not null,
  Description varchar(57) not null,
  Founded year not null
);

CREATE TABLE cw_2.Industry (
  Industry_ID integer primary key,
  Industry varchar(52) not null unique
);

CREATE TABLE cw_2.Employee (
  User_ID char(15) primary key,
  First_Name varchar(10) not null,
  Last_Name varchar(10) not null,
  Sex char(6) not null,
  Birth_Date date not null
);

CREATE TABLE cw_2.Contact_Information (
  Contact_ID char(20) primary key,
  Email varchar(30) not null unique,
  Phone char(17) not null unique
);

create table cw_2.Role (
Role_ID char(9) primary key,
Role_Title varchar(59) not null,
Salary integer not null,
Experience_Levels char(12) not null
);

create table cw_2.Assign (
Assign_Number integer primary key,
Organization_ID char(15) not null,
Industry_ID integer not null,
User_ID char(15) not null,
Contact_ID char(20) not null,
Role_ID char(9) not null,
foreign key (Organization_ID) references cw_2.Organization(Organization_ID),
foreign key (Industry_ID) references cw_2.Industry(Industry_ID),
foreign key (User_ID) references cw_2.Employee(User_ID),
foreign key (Contact_ID) references cw_2.Contact_Information(Contact_ID),
foreign key (Role_ID) references cw_2.Role(Role_ID)
);
```

202118010418



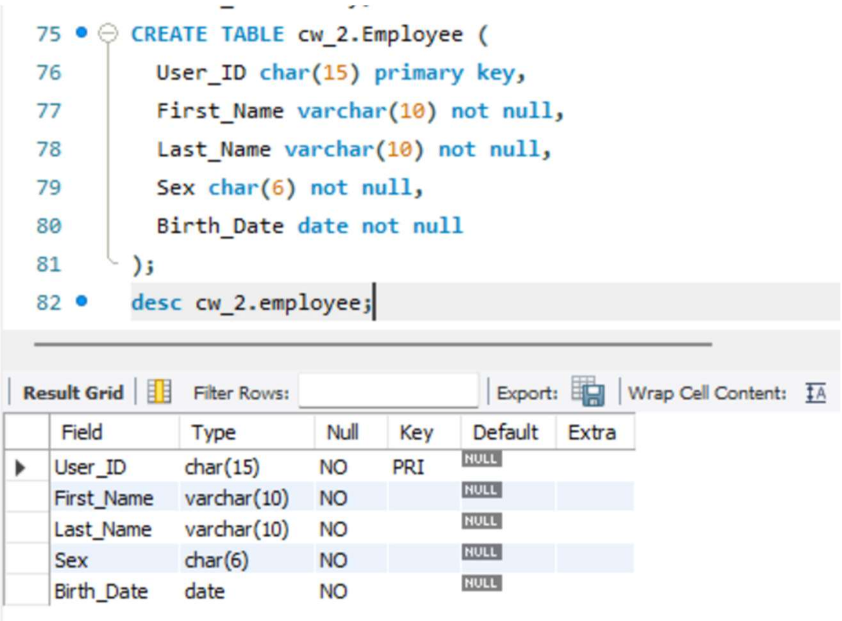
```sql
CREATE TABLE cw_2.Organization (
  Organization_ID char(15) primary key,
  Name varchar(31) not null,
  Website varchar(34) not null unique,
  Country varchar(33) not null,
  Description varchar(57) not null,
  Founded year not null
);
desc cw_2.organization;
```

Field	Type	Null	Key	Default	Extra
Organization_ID	char(15)	NO	PRI	NULL	
Name	varchar(31)	NO		NULL	
Website	varchar(34)	NO	UNI	NULL	
Country	varchar(33)	NO		NULL	
Description	varchar(57)	NO		NULL	
Founded	year	NO		NULL	



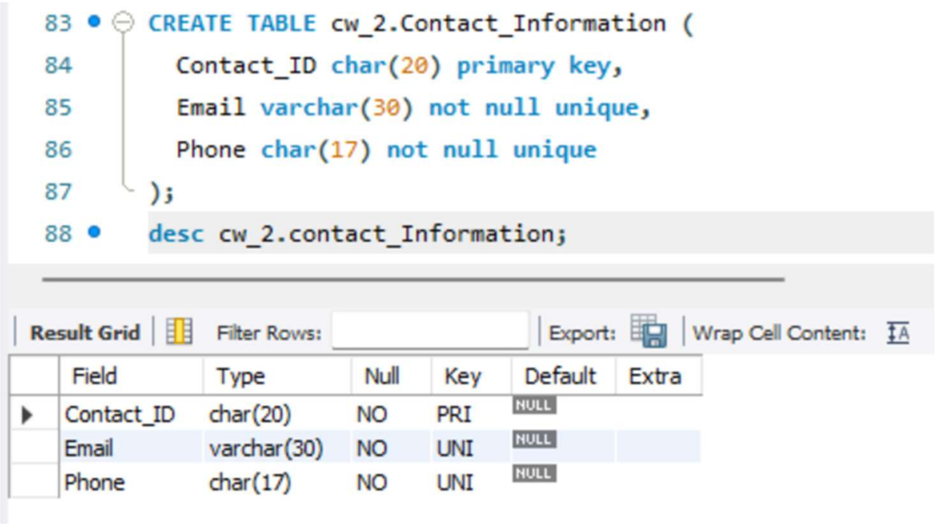
```sql
CREATE TABLE cw_2.Industry (
  Industry_ID integer primary key,
  Industry varchar(52) not null unique
);
desc cw_2.industry;
```

Field	Type	Null	Key	Default	Extra
Industry_ID	int	NO	PRI	NULL	
Industry	varchar(52)	NO	UNI	NULL	



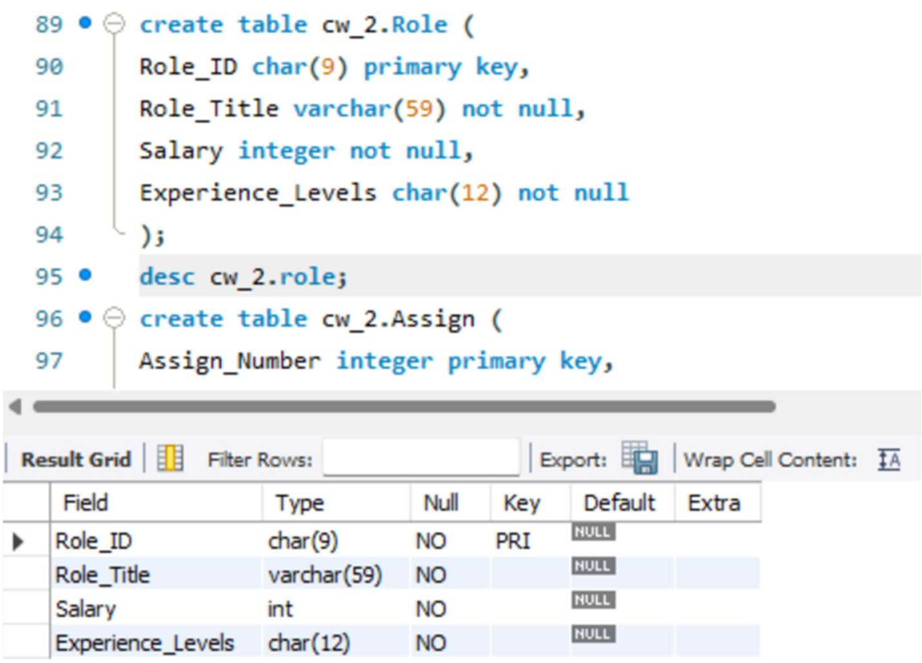
```
CREATE TABLE cw_2.Employee (
  User_ID char(15) primary key,
  First_Name varchar(10) not null,
  Last_Name varchar(10) not null,
  Sex char(6) not null,
  Birth_Date date not null
);
desc cw_2.employee;
```

Field	Type	Null	Key	Default	Extra
User_ID	char(15)	NO	PRI	NULL	
First_Name	varchar(10)	NO		NULL	
Last_Name	varchar(10)	NO		NULL	
Sex	char(6)	NO		NULL	
Birth_Date	date	NO		NULL	



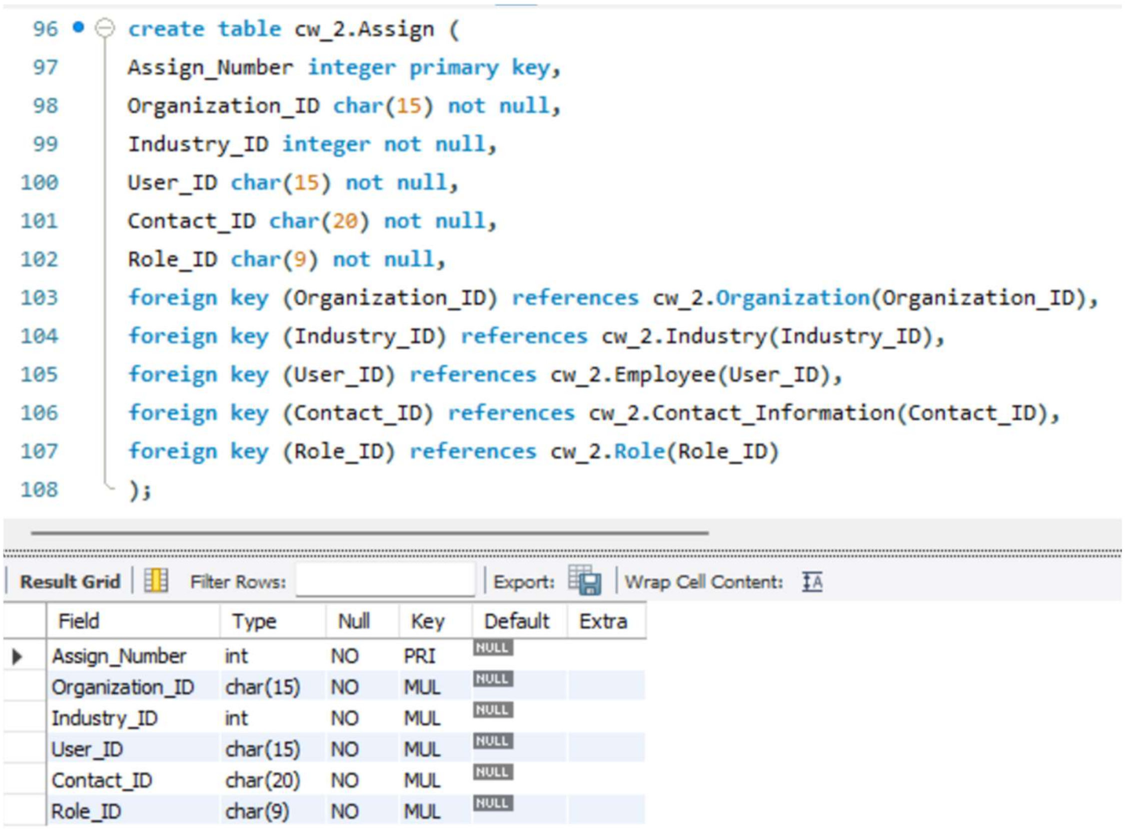
```
CREATE TABLE cw_2.Contact_Information (
  Contact_ID char(20) primary key,
  Email varchar(30) not null unique,
  Phone char(17) not null unique
);
desc cw_2.contact_Information;
```

Field	Type	Null	Key	Default	Extra
Contact_ID	char(20)	NO	PRI	NULL	
Email	varchar(30)	NO	UNI	NULL	
Phone	char(17)	NO	UNI	NULL	



```
create table cw_2.Role (
Role_ID char(9) primary key,
Role_Title varchar(59) not null,
Salary integer not null,
Experience_Levels char(12) not null
);
desc cw_2.role;
create table cw_2.Assign (
Assign_Number integer primary key,
```

Field	Type	Null	Key	Default	Extra
Role_ID	char(9)	NO	PRI	NULL	
Role_Title	varchar(59)	NO		NULL	
Salary	int	NO		NULL	
Experience_Levels	char(12)	NO		NULL	



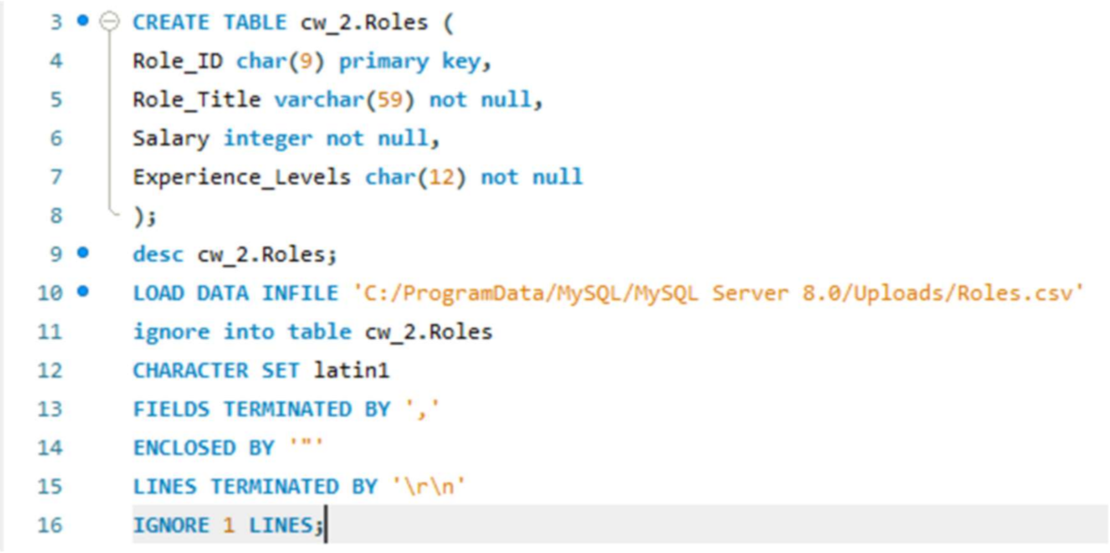
```
create table cw_2.Assign (
Assign_Number integer primary key,
Organization_ID char(15) not null,
Industry_ID integer not null,
User_ID char(15) not null,
Contact_ID char(20) not null,
Role_ID char(9) not null,
foreign key (Organization_ID) references cw_2.Organization(Organization_ID),
foreign key (Industry_ID) references cw_2.Industry(Industry_ID),
foreign key (User_ID) references cw_2.Employee(User_ID),
foreign key (Contact_ID) references cw_2.Contact_Information(Contact_ID),
foreign key (Role_ID) references cw_2.Role(Role_ID)
);
```

Field	Type	Null	Key	Default	Extra
Assign_Number	int	NO	PRI	NULL	
Organization_ID	char(15)	NO	MUL	NULL	
Industry_ID	int	NO	MUL	NULL	
User_ID	char(15)	NO	MUL	NULL	
Contact_ID	char(20)	NO	MUL	NULL	
Role_ID	char(9)	NO	MUL	NULL	

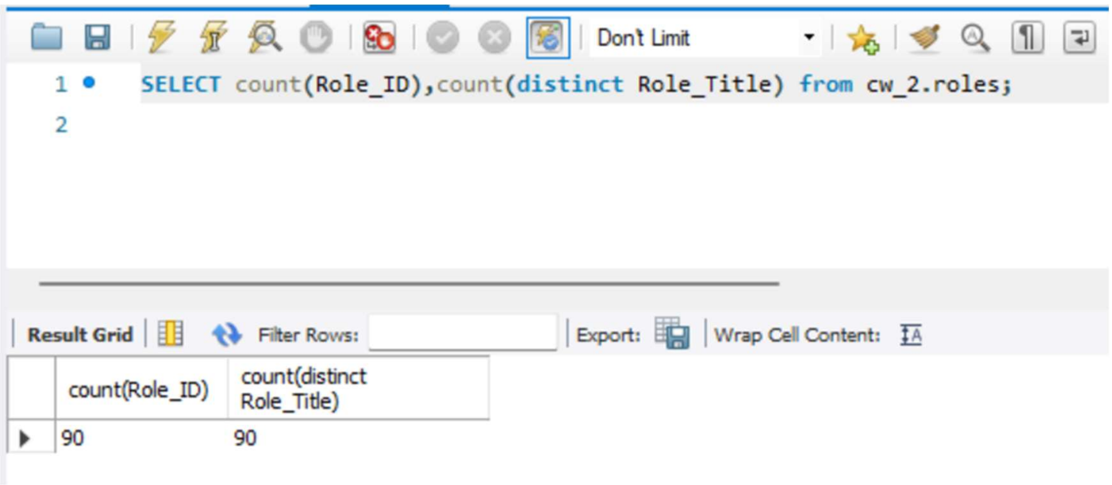
6) Ensure that the data is clean, and describe the steps taken. If you need to remove some of it, explain why that is the case.

Answer:

1. Check the three tables and find duplicate records in Roles. If all of them are loaded, the primary key cannot be set (because the primary key requires uniqueness). In addition, the Job_Title field in Employees is duplicated with the Role_Title field in Roles, resulting in data redundancy.
2. Delete the duplicate record in the Roles table and set Role_ID as the primary key. Duplicate records are unreasonable, which may cause data inconsistency, waste storage space, and impair query performance. In addition, setting the primary key can ensure data integrity and consistency, and improve query efficiency. The implementation method is as follows:
   i. Set Role_ID as the primary key in the Roles table.
   ii. Use 'ignore into' to filter out duplicate records when loading data.



```
CREATE TABLE cw_2.Roles (
Role_ID char(9) primary key,
Role_Title varchar(59) not null,
Salary integer not null,
Experience_Levels char(12) not null
);
desc cw_2.Roles;
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Roles.csv'
ignore into table cw_2.Roles
CHARACTER SET latin1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

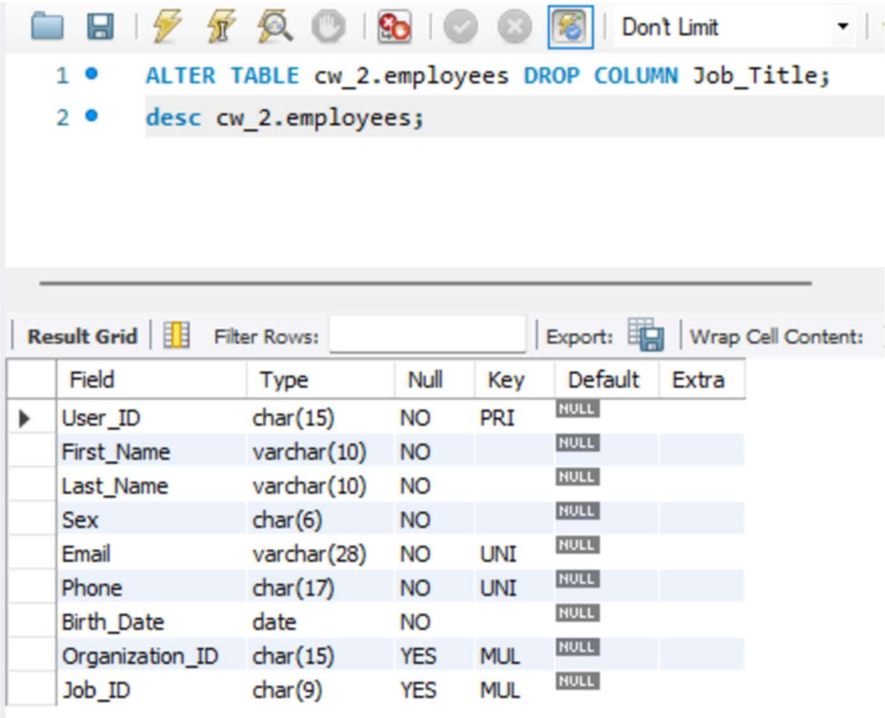


```
SELECT count(Role_ID),count(distinct Role_Title) from cw_2.roles;
```

count(Role_ID)	count(distinct Role_Title)
90	90

3. Delete Job_Title from Employees. The Employees and Roles tables are associated with Job_ID as the foreign key and Role_ID as the primary key, respectively. Therefore, the Job_Title field in Employees is the same as the Role_Title field in Roles, resulting in redundant data. Deleting the Job_Title field of Employees can save storage space and improve query efficiency. The implementation method is as follows:

i. Use 'ALTER TABLE cw_2.Employees DROP COLUMN Job_Title;' to delete it.



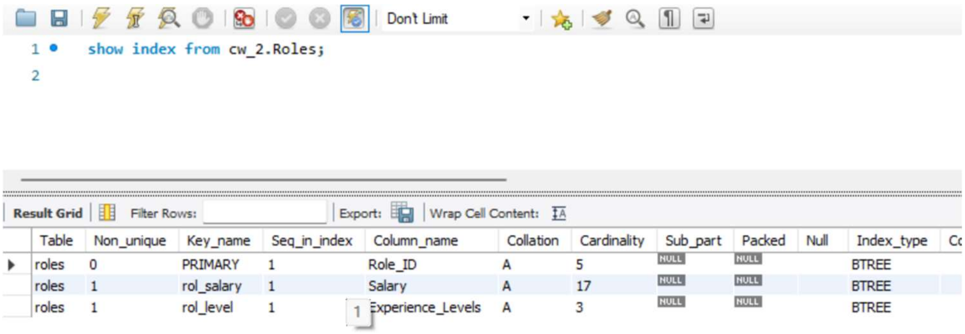
```
ALTER TABLE cw_2.employees DROP COLUMN Job_Title;
desc cw_2.employees;
```

Field	Type	Null	Key	Default	Extra
User_ID	char(15)	NO	PRI	NULL	
First_Name	varchar(10)	NO		NULL	
Last_Name	varchar(10)	NO		NULL	
Sex	char(6)	NO		NULL	
Email	varchar(28)	NO	UNI	NULL	
Phone	char(17)	NO	UNI	NULL	
Birth_Date	date	NO		NULL	
Organization_ID	char(15)	YES	MUL	NULL	
Job_ID	char(9)	YES	MUL	NULL	

7) Identify appropriate columns and create necessary indexes to optimize query performance.

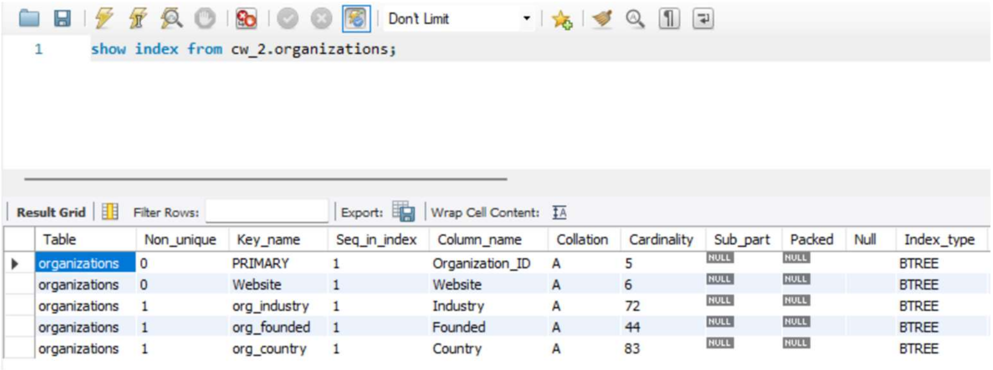
Answer:

create index rol_salary on cw_2.roles(salary);
create index rol_level on cw_2.roles(experience_levels);
create index org_industry on cw_2.organizations(industry);
create index org_founded on cw_2.organizations(founded);
create index org_country on cw_2.organizations(country);
create index emp_firstName on cw_2.employees(First_Name);
create index emp_birth on cw_2.employees(birth_date);
create index emp_sex on cw_2.employees(sex);



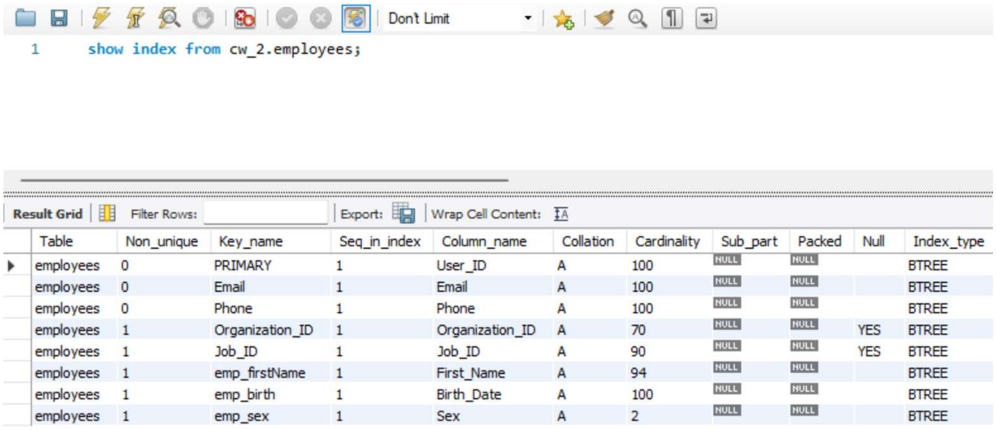
```
show index from cw_2.Roles;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Co
roles	0	PRIMARY	1	Role_ID	A	5	NULL	NULL		BTREE	
roles	1	rol_salary	1	Salary	A	17	NULL	NULL		BTREE	
roles	1	rol_level	1	Experience_Levels	A	3	NULL	NULL		BTREE	



```
show index from cw_2.organizations;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
organizations	0	PRIMARY	1	Organization_ID	A	5	NULL	NULL		BTREE
organizations	0	Website	1	Website	A	6	NULL	NULL		BTREE
organizations	1	org_industry	1	Industry	A	72	NULL	NULL		BTREE
organizations	1	org_founded	1	Founded	A	44	NULL	NULL		BTREE
organizations	1	org_country	1	Country	A	83	NULL	NULL		BTREE



```
show index from cw_2.employees;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
employees	0	PRIMARY	1	User_ID	A	100	NULL	NULL		BTREE
employees	0	Email	1	Email	A	100	NULL	NULL		BTREE
employees	0	Phone	1	Phone	A	100	NULL	NULL		BTREE
employees	1	Organization_ID	1	Organization_ID	A	70	NULL	NULL	YES	BTREE
employees	1	Job_ID	1	Job_ID	A	90	NULL	NULL	YES	BTREE
employees	1	emp_firstName	1	First_Name	A	94	NULL	NULL		BTREE
employees	1	emp_birth	1	Birth_Date	A	100	NULL	NULL		BTREE
employees	1	emp_sex	1	Sex	A	2	NULL	NULL		BTREE

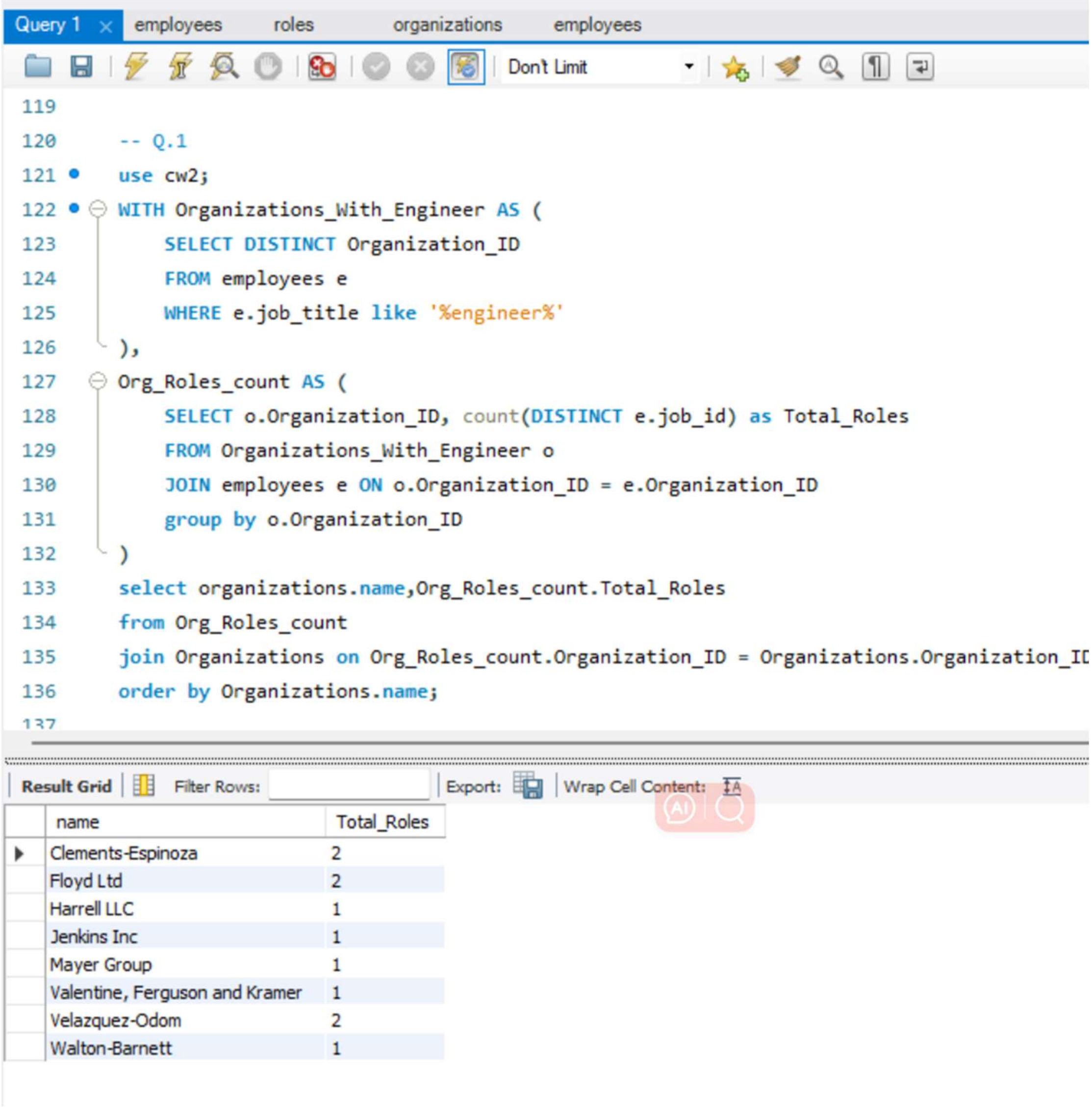
8) Answer the following questions using database queries and include your SQL statements.

Answer:

1.

```
use cw2;
WITH Organizations_With_Engineer AS (
    SELECT DISTINCT Organization_ID
    FROM employees e
    WHERE e.job_title like '%engineer%'
),
Org_Roles_count AS (
    SELECT o.Organization_ID, count(DISTINCT e.job_id) as Total_Roles
    FROM Organizations_With_Engineer o
    JOIN employees e ON o.Organization_ID = e.Organization_ID
    group by o.Organization_ID
)
select organizations.name,Org_Roles_count.Total_Roles
from Org_Roles_count
join Organizations on Org_Roles_count.Organization_ID =
Organizations.Organization_ID
```

```
order by Organizations.name;
```

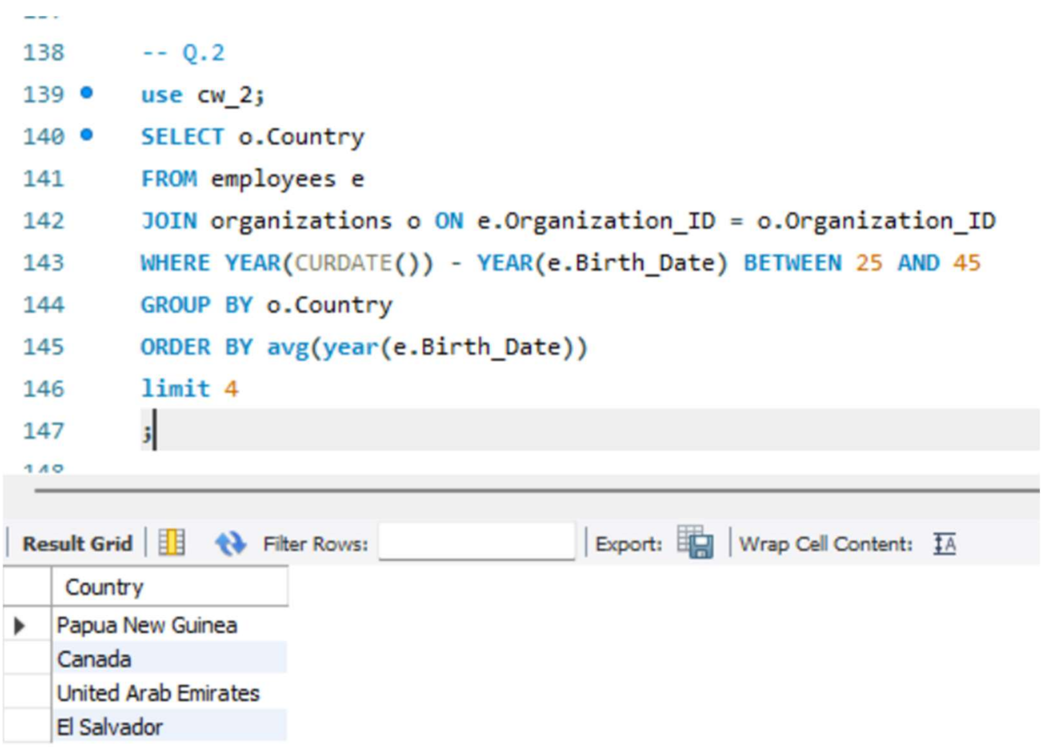


```

-- Q.1
use cw2;
WITH Organizations_With_Engineer AS (
    SELECT DISTINCT Organization_ID
    FROM employees e
    WHERE e.job_title like '%engineer%'
),
Org_Roles_count AS (
    SELECT o.Organization_ID, count(DISTINCT e.job_id) as Total_Roles
    FROM Organizations_With_Engineer o
    JOIN employees e ON o.Organization_ID = e.Organization_ID
    group by o.Organization_ID
)
select organizations.name,Org_Roles_count.Total_Roles
from Org_Roles_count
join Organizations on Org_Roles_count.Organization_ID = Organizations.Organization_ID
order by Organizations.name;
```

name	Total_Roles
Clements-Espinoza	2
Floyd Ltd	2
Harrell LLC	1
Jenkins Inc	1
Mayer Group	1
Valentine, Ferguson and Kramer	1
Velazquez-Odom	2
Walton-Barnett	1

2.

```
use cw_2;
SELECT o.Country
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE YEAR(CURDATE()) - YEAR(e.Birth_Date) BETWEEN 25 AND 45
GROUP BY o.Country
ORDER BY avg(year(e.Birth_Date))
limit 4;
```

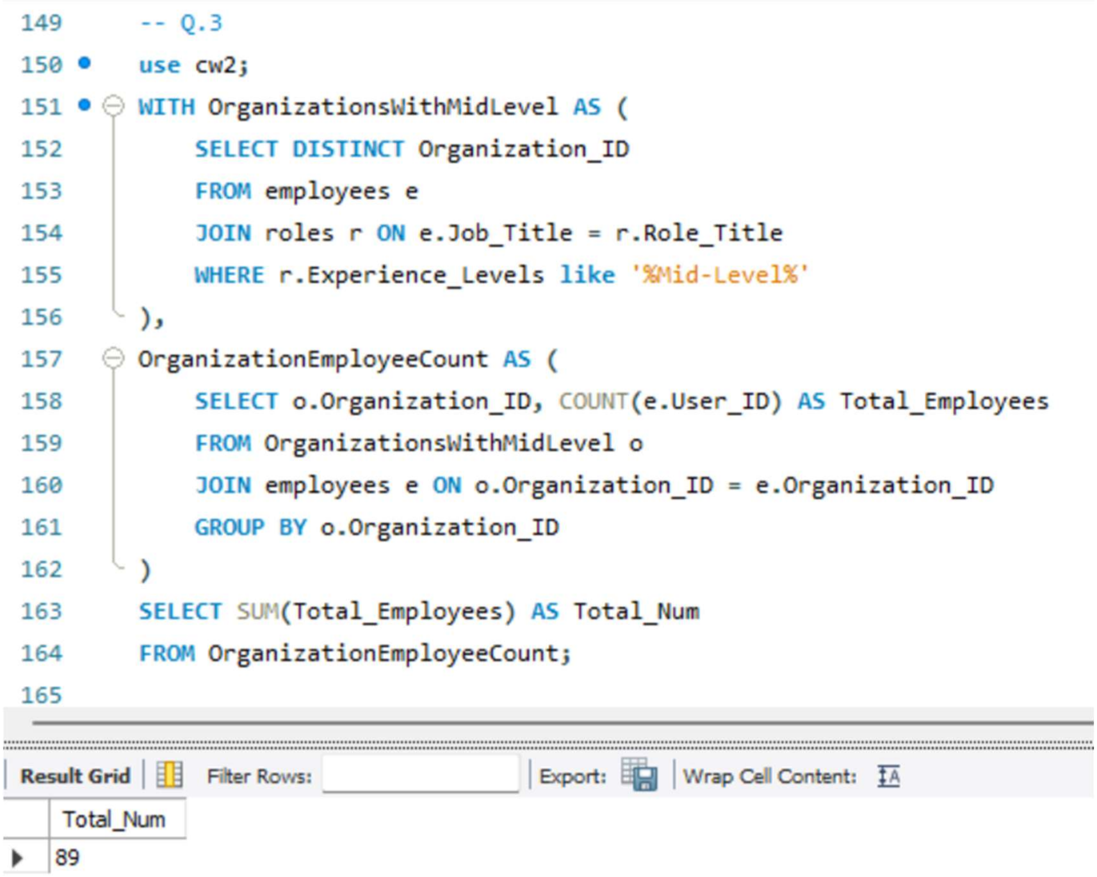


```
-- Q.2
use cw_2;
SELECT o.Country
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE YEAR(CURDATE()) - YEAR(e.Birth_Date) BETWEEN 25 AND 45
GROUP BY o.Country
ORDER BY avg(year(e.Birth_Date))
limit 4
;
```

Country
Papua New Guinea
Canada
United Arab Emirates
El Salvador

3.

```
use cw2;
WITH OrganizationsWithMidLevel AS (
    SELECT DISTINCT Organization_ID
    FROM employees e
    JOIN roles r ON e.Job_Title = r.Role_Title
    WHERE r.Experience_Levels like '%Mid-Level%'
),
OrganizationEmployeeCount AS (
    SELECT o.Organization_ID, COUNT(e.User_ID) AS Total_Employees
    FROM OrganizationsWithMidLevel o
    JOIN employees e ON o.Organization_ID = e.Organization_ID
    GROUP BY o.Organization_ID
)
SELECT SUM(Total_Employees) AS Total_Num
FROM OrganizationEmployeeCount;
```

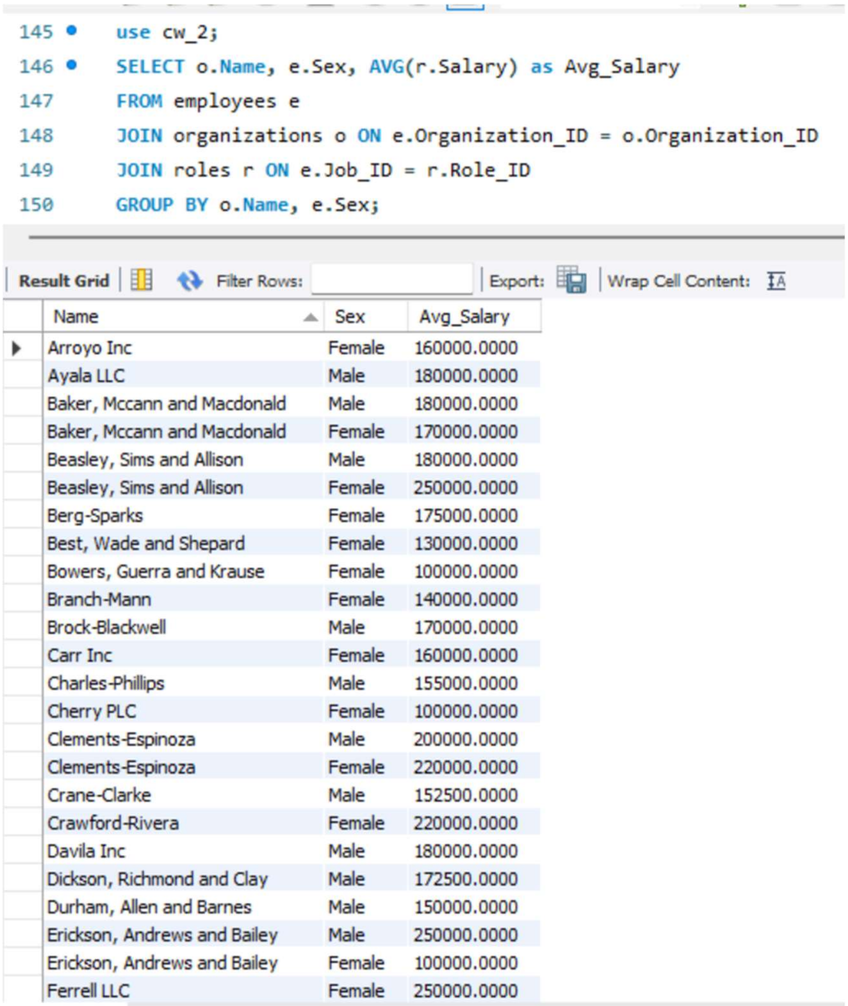


```
-- Q.3
use cw2;
WITH OrganizationsWithMidLevel AS (
    SELECT DISTINCT Organization_ID
    FROM employees e
    JOIN roles r ON e.Job_Title = r.Role_Title
    WHERE r.Experience_Levels like '%Mid-Level%'
),
OrganizationEmployeeCount AS (
    SELECT o.Organization_ID, COUNT(e.User_ID) AS Total_Employees
    FROM OrganizationsWithMidLevel o
    JOIN employees e ON o.Organization_ID = e.Organization_ID
    GROUP BY o.Organization_ID
)
SELECT SUM(Total_Employees) AS Total_Num
FROM OrganizationEmployeeCount;
```

```
Total_Num
89
```

4.

```
use cw_2;
SELECT o.Name, e.Sex, AVG(r.Salary) as Avg_Salary
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
JOIN roles r ON e.Job_ID = r.Role_ID
GROUP BY o.Name, e.Sex;
```

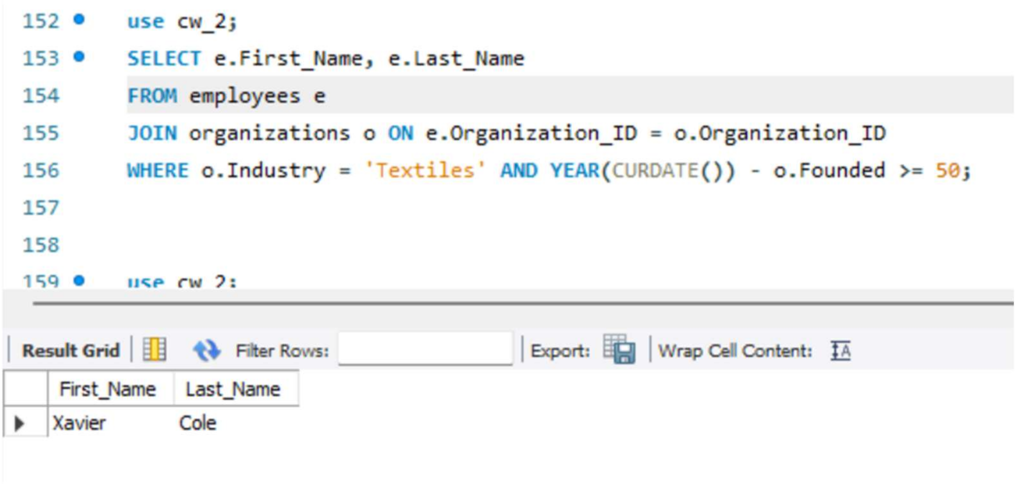


```
use cw_2;
SELECT o.Name, e.Sex, AVG(r.Salary) as Avg_Salary
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
JOIN roles r ON e.Job_ID = r.Role_ID
GROUP BY o.Name, e.Sex;
```

Name	Sex	Avg_Salary
Arroyo Inc	Female	160000.0000
Ayala LLC	Male	180000.0000
Baker, Mccann and Macdonald	Male	180000.0000
Baker, Mccann and Macdonald	Female	170000.0000
Beasley, Sims and Allison	Male	180000.0000
Beasley, Sims and Allison	Female	250000.0000
Berg-Sparks	Female	175000.0000
Best, Wade and Shepard	Female	130000.0000
Bowers, Guerra and Krause	Female	100000.0000
Branch-Mann	Female	140000.0000
Brock-Blackwell	Male	170000.0000
Carr Inc	Female	160000.0000
Charles-Phillips	Male	155000.0000
Cherry PLC	Female	100000.0000
Clements-Espinoza	Male	200000.0000
Clements-Espinoza	Female	220000.0000
Crane-Clarke	Male	152500.0000
Crawford-Rivera	Female	220000.0000
Davila Inc	Male	180000.0000
Dickson, Richmond and Clay	Male	172500.0000
Durham, Allen and Barnes	Male	150000.0000
Erickson, Andrews and Bailey	Male	250000.0000
Erickson, Andrews and Bailey	Female	100000.0000
Ferrell LLC	Female	250000.0000

5.

use cw_2;
SELECT e.First_Name, e.Last_Name
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE o.Industry = 'Textiles' AND YEAR(CURDATE()) - o.Founded >= 50;

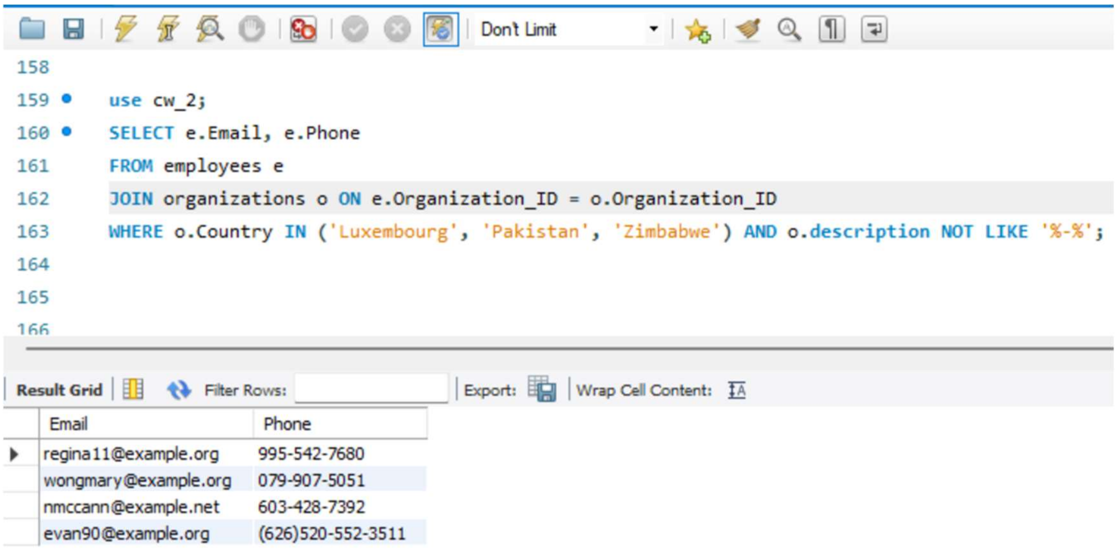


```
use cw_2;
SELECT e.First_Name, e.Last_Name
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE o.Industry = 'Textiles' AND YEAR(CURDATE()) - o.Founded >= 50;


use cw_2;
```

First_Name	Last_Name
Xavier	Cole

6.

use cw_2;
SELECT e.Email, e.Phone
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE o.Country IN ('Luxembourg', 'Pakistan', 'Zimbabwe') AND o.description NOT LIKE '%-%';



```
use cw_2;
SELECT e.Email, e.Phone
FROM employees e
JOIN organizations o ON e.Organization_ID = o.Organization_ID
WHERE o.Country IN ('Luxembourg', 'Pakistan', 'Zimbabwe') AND o.description NOT LIKE '%-%';
```

Email	Phone
regina11@example.org	995-542-7680
wongmary@example.org	079-907-5051
nmccann@example.net	603-428-7392
evan90@example.org	(626)520-552-3511

9) Artificial Intelligence (AI) plays a transformative role in database management for modern organizations, enabling them to gain valuable insights, make data-driven decisions, and identify innovation opportunities. Discuss the opportunities and challenges of AI-driven database administration, employing AI technologies like machine learning (ML) and natural language processing (NLP), considering crucial factors such as efficiency, reliability, emerging trends, and ethical concerns. Include relevant references to support your discussion. (Maximum 1000 words, including the references).

Answer:

Introduction

Artificial Intelligence (AI) is a branch of computer science that uses machine learning (ML) to achieve human intelligence tasks. Natural Language Processing (NLP) is a subfield of AI that uses ML to understand and generate natural language. Driving database management with artificial intelligence, such as machine learning (ML) and natural language processing (NLP), can bring many benefits to database management as well as many challenges. The advantage of AI-driven database management is that it can improve the efficiency of database operations, enhance the reliability of database systems, and facilitate data-driven decision making. However, AI-driven database management also faces challenges in terms of efficiency, reliability, emerging trends, and ethics.

Opportunity

Efficiency:

Firstly, AI technology improves the efficiency of database management by automating and optimizing queries. Ai-driven database management can improve the efficiency of database operations by automating routine tasks, optimizing query execution, and reducing human errors. For example, AI can automatically select the most appropriate indexing, partitioning, and compression strategy based on the characteristics and distribution of the data, thereby improving the speed and accuracy of the query [1]. AI can also automatically allocate resources, adjust parameters, and balance performance based on database workload [2]. In addition, AI can allow users to ask queries in natural language through natural language interfaces without having to master complex query languages [3].

Reliability:

Secondly, AI technology improves the reliability of database systems through predictive maintenance and real-time monitoring. Ai-driven database management can enhance the reliability of database systems by detecting and resolving performance issues, ensuring data quality, and providing backup and recovery mechanisms. For example, AI can monitor the status of databases in real time, detect and predict failures, and provide repair suggestions [4]. AI can also clean, validate, and transform data to eliminate inconsistencies, duplicates, and errors. Furthermore, AI can automatically select the best backup and recovery strategy to prevent data loss and corruption based on its importance, sensitivity, and frequency of change.

Trend identification:

Thirdly, AI technologies help organizations identify and leverage emerging trends to drive innovation. Ai-driven database management can enable data-driven decision making by analyzing complex data patterns, providing valuable insights, and facilitating data exploration and visualization. For example, AI can use machine learning algorithms to extract meaningful information from large amounts of data, discover hidden associations, and predict future trends. AI can also use natural language processing techniques to extract useful knowledge from unstructured data, understand the semantics of the data, and generate natural language reports. In addition, AI can utilize data visualization tools to present data in the form of graphs, charts, and dashboards to help users understand and explore the data more intuitively.

Challenge

However, AI-driven database management also faces several challenges that we need to seriously consider and address.

Efficiency:

Implementing AI techniques may increase system complexity and resource consumption, leading to decreased efficiency. For example, AI algorithms and models need to process large amounts of data, perform complex calculations, and store large numbers of parameters, which can lead to degraded performance, increased latency, and increased cost of databases. Therefore, we need to optimize the design, selection, and deployment of AI algorithms and models to improve database efficiency and save resources.

Reliability:

Second, AI-driven database management faces reliability challenges, such as the robustness, accuracy, and consistency of AI output and behavior. For example, AI output and behavior can be affected by the quality, distribution, and noise of the data, leading to errors, bias, and uncertainty. Therefore, we need to verify and evaluate the correctness, rationality, and predictability of the AI output and behavior to improve the reliability and trust of the database.

Sustainability:

Ensuring the sustainability of solutions is a challenge given the rapid evolution of AI technologies. For example, with the development of the Internet, Internet of things, and social media, the types, sources, and formats of data are becoming more diverse, complex, and dynamic, which requires AI to be able to handle different data structures, semantics, and languages. At the same time, with the progress of AI technology, AI can achieve more functions, such as self-learning, self-adaptation, and self-optimization, which requires AI to be able to better cooperate and coordinate with database systems. Therefore, we need to constantly update and extend the architectures, methods, and standards for AI-driven database management to accommodate new data and AI developments.

Ethical Issues:

Ethical issues such as privacy and bias can be raised when using AI to process sensitive data. For example, AI may collect, analyze, and share users' sensitive data, such as personal information, health status, and consumption behavior, which may violate users' privacy rights, cause data leakage, and trigger data misuse. AI may also make unfair, inaccurate, and unfriendly judgments about data and users, such as discrimination, bias, and misdirection, which may harm users' interests, trust, and satisfaction. Therefore, we need to develop and adhere to ethical principles, norms, and responsibilities for AI-driven database management to protect the rights, respect, and dignity of data and users.

Conclusion

In summary, AI-driven database management is a promising and challenging field that can not only bring many benefits to modern organizations and society, such as aiding data management, improving efficiency and data analysis, but also faces many challenges, such as sustainability and ethical issues, etc. Therefore, this requires more efforts, such as developing appropriate ethical guidelines for AI and focusing on balancing innovation and risk management. It is believed that AI-driven database management technology will surely contribute to mankind in the future.

Reference

1. Scribbr. “The Beginner's Guide to Writing an Essay | Steps & Examples.” Scribbr. https://www.scribbr.com/category/academic-essay/ . Accessed in: 12/17/2023.

2. MoversBoost Integrates with ChatGPT for a 50,000 AI SEO Database for Movers [J]. M2 Presswire, 2023,

3. Deepen AI and WMG, University of Warwick Launch Safety Pool Scenario Database for Automated Driv

4. Excelra Knowledge Solutions Pvt Ltd; Cyclica to use Excelra's GOSTAR Database to develop AI &amp; ML based Deep Learning Algorithm for Drug Target Identification [J]. Computer Technology Journal, 2020, 16-.