Managing Employee Data in MySQL

Shweta Ajay Shinde

Masters in Data Analytics, San Jose State University

Data 226: Data warehousing

Instructor: Keeyong Han

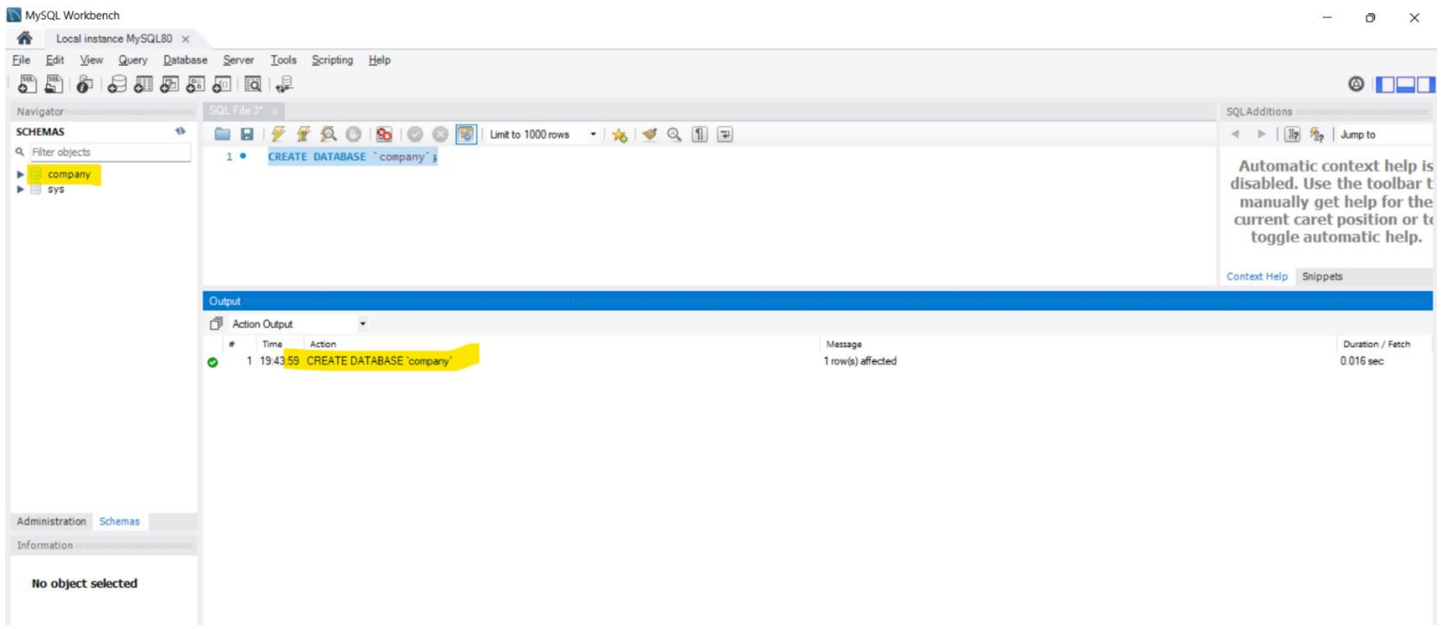
2nd Sept 2024

Managing Employee Data in MySQL

Task - 1 : Database Creation & Schema Design

1. **Create a new MySQL database named 'company'.**

   SQL:- CREATE DATABASE `company`;



```
CREATE DATABASE `company`;
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

```
#   Time       Action                       Message             Duration / Fetch
1   19:43:59   CREATE DATABASE 'company'    1 row(s) affected   0.016 sec
```

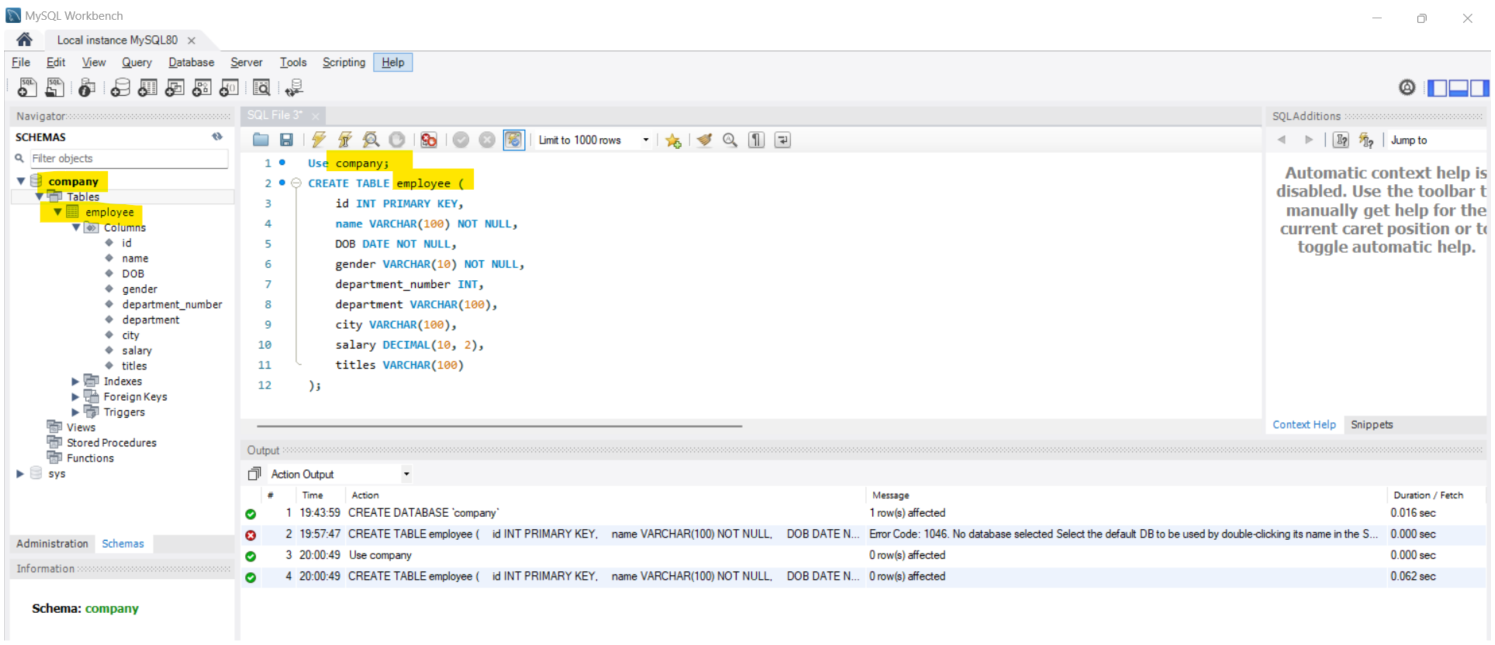
No object selected

2. **Create table employee with schema (id, name, DOB, gender, department number, department, city, salary, titles).**

   SQL:- Use company;

```
CREATE TABLE employee (
    id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    DOB DATE NOT NULL,
    gender VARCHAR(10) NOT NULL,
    department_number INT,
    department VARCHAR(100),
    city VARCHAR(100),
    salary DECIMAL(10, 2),
    titles VARCHAR(100)
);
```

Managing Employee Data in MySQL



```
Use company;
CREATE TABLE employee (
    id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    DOB DATE NOT NULL,
    gender VARCHAR(10) NOT NULL,
    department_number INT,
    department VARCHAR(100),
    city VARCHAR(100),
    salary DECIMAL(10, 2),
    titles VARCHAR(100)
);
```

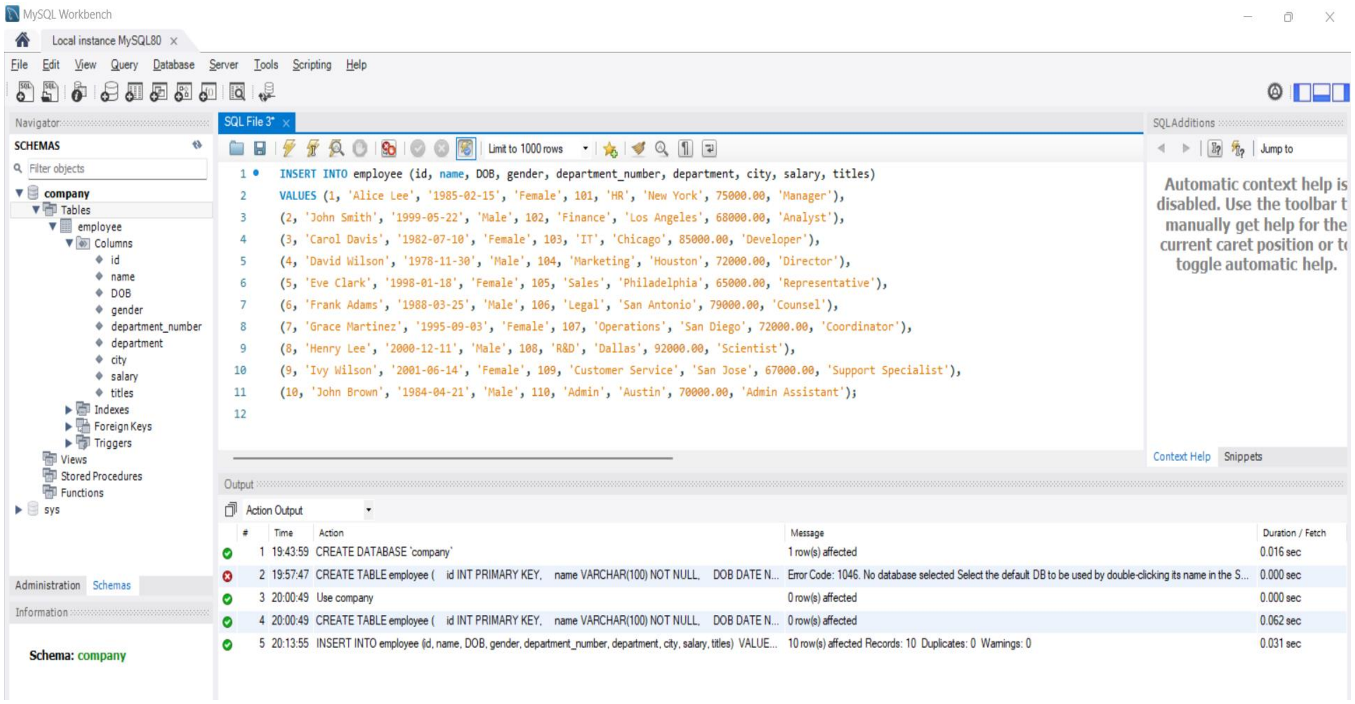
```
#  Time      Action                                                                             Message                                                                                  Duration / Fetch
1  19:43:59  CREATE DATABASE 'company'                                                          1 row(s) affected                                                                        0.016 sec
2  19:57:47  CREATE TABLE employee (   id INT PRIMARY KEY,   name VARCHAR(100) NOT NULL,   DOB DATE N...   Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the S...   0.000 sec
3  20:00:49  Use company                                                                        0 row(s) affected                                                                        0.000 sec
4  20:00:49  CREATE TABLE employee (   id INT PRIMARY KEY,   name VARCHAR(100) NOT NULL,   DOB DATE N...   0 row(s) affected                                                                        0.062 sec
```

Schema: company

3. Insert 10 records to the employee table.

SQL:-
INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles)
VALUES (1, 'Alice Lee', '1985-02-15', 'Female', 101, 'HR', 'New York', 75000.00, 'Manager'),
(2, 'John Smith', '1999-05-22', 'Male', 102, 'Finance', 'Los Angeles', 68000.00, 'Analyst'),
(3, 'Carol Davis', '1982-07-10', 'Female', 103, 'IT', 'Chicago', 85000.00, 'Developer'),
(4, 'David Wilson', '1978-11-30', 'Male', 104, 'Marketing', 'Houston', 72000.00, 'Director'),
(5, 'Eve Clark', '1998-01-18', 'Female', 105, 'Sales', 'Philadelphia', 65000.00, 'Representative'),
(6, 'Frank Adams', '1988-03-25', 'Male', 106, 'Legal', 'San Antonio', 79000.00, 'Counsel'),
(7, 'Grace Martinez', '1995-09-03', 'Female', 107, 'Operations', 'San Diego', 72000.00, 'Coordinator'),
(8, 'Henry Lee', '2000-12-11', 'Male', 108, 'R&D', 'Dallas', 92000.00, 'Scientist'),
(9, 'Ivy Wilson', '2001-06-14', 'Female', 109, 'Customer Service', 'San Jose', 67000.00, 'Support Specialist'),
(10, 'John Brown', '1984-04-21', 'Male', 110, 'Admin', 'Austin', 70000.00, 'Admin Assistant');

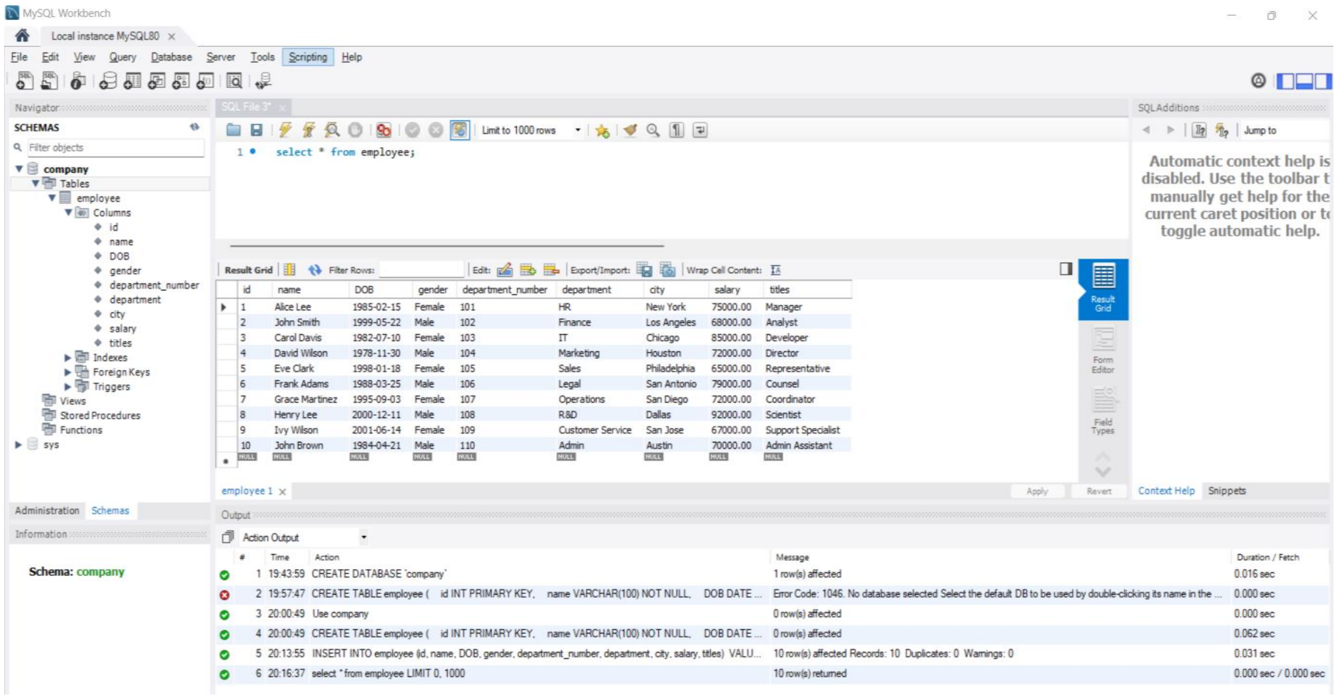
Managing Employee Data in MySQL



```sql
INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles)
VALUES (1, 'Alice Lee', '1985-02-15', 'Female', 101, 'HR', 'New York', 75000.00, 'Manager'),
(2, 'John Smith', '1999-05-22', 'Male', 102, 'Finance', 'Los Angeles', 68000.00, 'Analyst'),
(3, 'Carol Davis', '1982-07-10', 'Female', 103, 'IT', 'Chicago', 85000.00, 'Developer'),
(4, 'David Wilson', '1978-11-30', 'Male', 104, 'Marketing', 'Houston', 72000.00, 'Director'),
(5, 'Eve Clark', '1998-01-18', 'Female', 105, 'Sales', 'Philadelphia', 65000.00, 'Representative'),
(6, 'Frank Adams', '1988-03-25', 'Male', 106, 'Legal', 'San Antonio', 79000.00, 'Counsel'),
(7, 'Grace Martinez', '1995-09-03', 'Female', 107, 'Operations', 'San Diego', 72000.00, 'Coordinator'),
(8, 'Henry Lee', '2000-12-11', 'Male', 108, 'R&D', 'Dallas', 92000.00, 'Scientist'),
(9, 'Ivy Wilson', '2001-06-14', 'Female', 109, 'Customer Service', 'San Jose', 67000.00, 'Support Specialist'),
(10, 'John Brown', '1984-04-21', 'Male', 110, 'Admin', 'Austin', 70000.00, 'Admin Assistant');
```

```
#  Time      Action                                                                                          Message                                                                                          Duration / Fetch
1  19:43:59  CREATE DATABASE 'company'                                                                       1 row(s) affected                                                                                0.016 sec
2  19:57:47  CREATE TABLE employee (    id INT PRIMARY KEY,    name VARCHAR(100) NOT NULL,    DOB DATE N...   Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the S...   0.000 sec
3  20:00:49  Use company                                                                                     0 row(s) affected                                                                                0.000 sec
4  20:00:49  CREATE TABLE employee (    id INT PRIMARY KEY,    name VARCHAR(100) NOT NULL,    DOB DATE N...   0 row(s) affected                                                                                0.062 sec
5  20:13:55  INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles)  VALUE...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0   0.031 sec
```

SQL:- Select * from employee



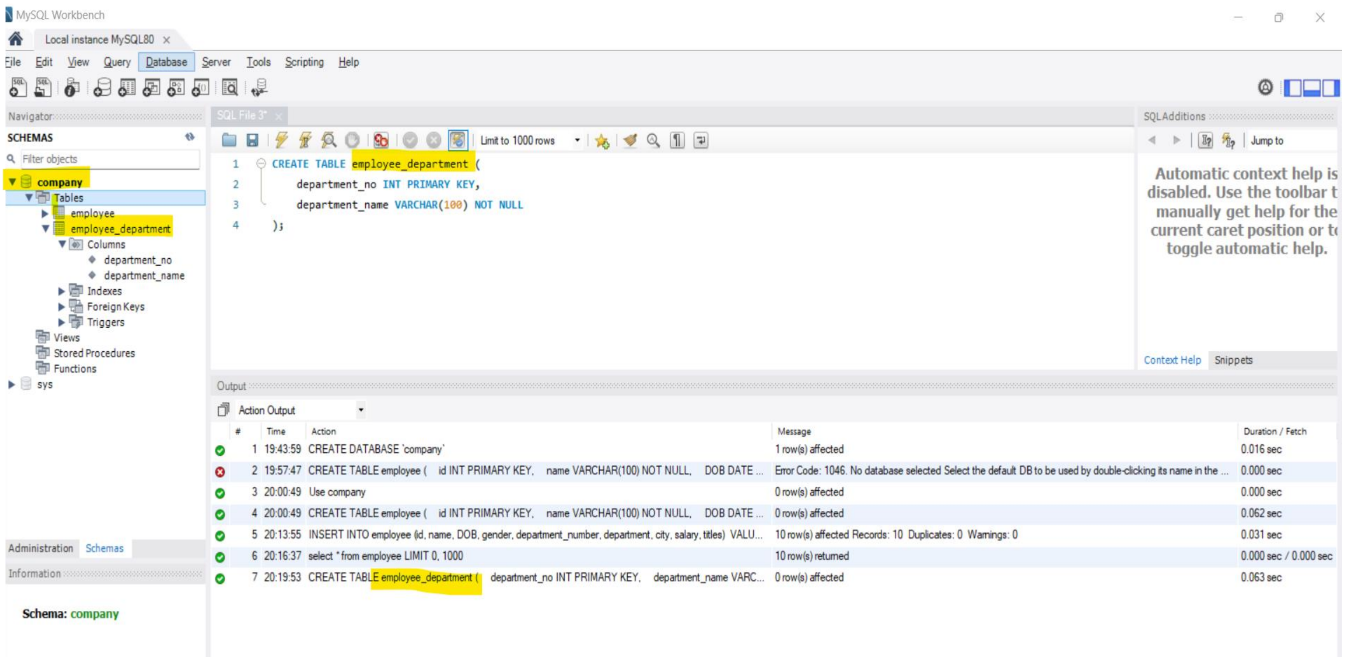
```sql
select * from employee;
```

id	name	DOB	gender	department_number	department	city	salary	titles
1	Alice Lee	1985-02-15	Female	101	HR	New York	75000.00	Manager
2	John Smith	1999-05-22	Male	102	Finance	Los Angeles	68000.00	Analyst
3	Carol Davis	1982-07-10	Female	103	IT	Chicago	85000.00	Developer
4	David Wilson	1978-11-30	Male	104	Marketing	Houston	72000.00	Director
5	Eve Clark	1998-01-18	Female	105	Sales	Philadelphia	65000.00	Representative
6	Frank Adams	1988-03-25	Male	106	Legal	San Antonio	79000.00	Counsel
7	Grace Martinez	1995-09-03	Female	107	Operations	San Diego	72000.00	Coordinator
8	Henry Lee	2000-12-11	Male	108	R&D	Dallas	92000.00	Scientist
9	Ivy Wilson	2001-06-14	Female	109	Customer Service	San Jose	67000.00	Support Specialist
10	John Brown	1984-04-21	Male	110	Admin	Austin	70000.00	Admin Assistant
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                                                                          Message                                                                                          Duration / Fetch
1  19:43:59  CREATE DATABASE 'company'                                                                       1 row(s) affected                                                                                0.016 sec
2  19:57:47  CREATE TABLE employee (    id INT PRIMARY KEY,    name VARCHAR(100) NOT NULL,    DOB DATE ...    Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the ...   0.000 sec
3  20:00:49  Use company                                                                                     0 row(s) affected                                                                                0.000 sec
4  20:00:49  CREATE TABLE employee (    id INT PRIMARY KEY,    name VARCHAR(100) NOT NULL,    DOB DATE ...    0 row(s) affected                                                                                0.062 sec
5  20:13:55  INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles)  VALU...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0   0.031 sec
6  20:16:37  select * from employee LIMIT 0, 1000                                                            10 row(s) returned                                                                               0.000 sec / 0.000 sec
```

4. **Create another table employee_department to store department information with Schema (department_no, department name).**

   SQL:-
   ```
   CREATE TABLE employee_department (
   department_no INT PRIMARY KEY,
   department_name VARCHAR(100) NOT NULL);
   ```



```
CREATE TABLE employee_department (
    department_no INT PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL
);
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

```
#  Time      Action                                                                                  Message                                                                                        Duration / Fetch
1  19:43:59  CREATE DATABASE 'company'                                                               1 row(s) affected                                                                              0.016 sec
2  19:57:47  CREATE TABLE employee (   id INT PRIMARY KEY,   name VARCHAR(100) NOT NULL,   DOB DATE ...  Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the ...  0.000 sec
3  20:00:49  Use company                                                                             0 row(s) affected                                                                              0.000 sec
4  20:00:49  CREATE TABLE employee (   id INT PRIMARY KEY,   name VARCHAR(100) NOT NULL,   DOB DATE ...  0 row(s) affected                                                                              0.062 sec
5  20:13:55  INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles) VALU...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0              0.031 sec
6  20:16:37  select * from employee LIMIT 0, 1000                                                    10 row(s) returned                                                                             0.000 sec / 0.000 sec
7  20:19:53  CREATE TABLE employee_department (   department_no INT PRIMARY KEY,   department_name VARC...  0 row(s) affected                                                                   0.063 sec
```

Schema: company

Write justification for using different datatypes for each attribute in 2 & 4.
Following is the justification of using different datatypes in the given tables.

employee Table

- **id** (INT), **department_number**(INT)
  This type is efficient for numeric identifiers and indexing. It ensures quick lookups and easy sorting.
- **name, gender, department**, **city** , **titles** (VARCHAR(100)):
  Varchar(100) handles variable-length data up to 100 characters. Usually, the data stored in these columns won't exceed 100 characters.
- **DOB** (DATE):
  The DATE data type stores dates in a consistent format. This makes it easy to perform calculations involving dates.

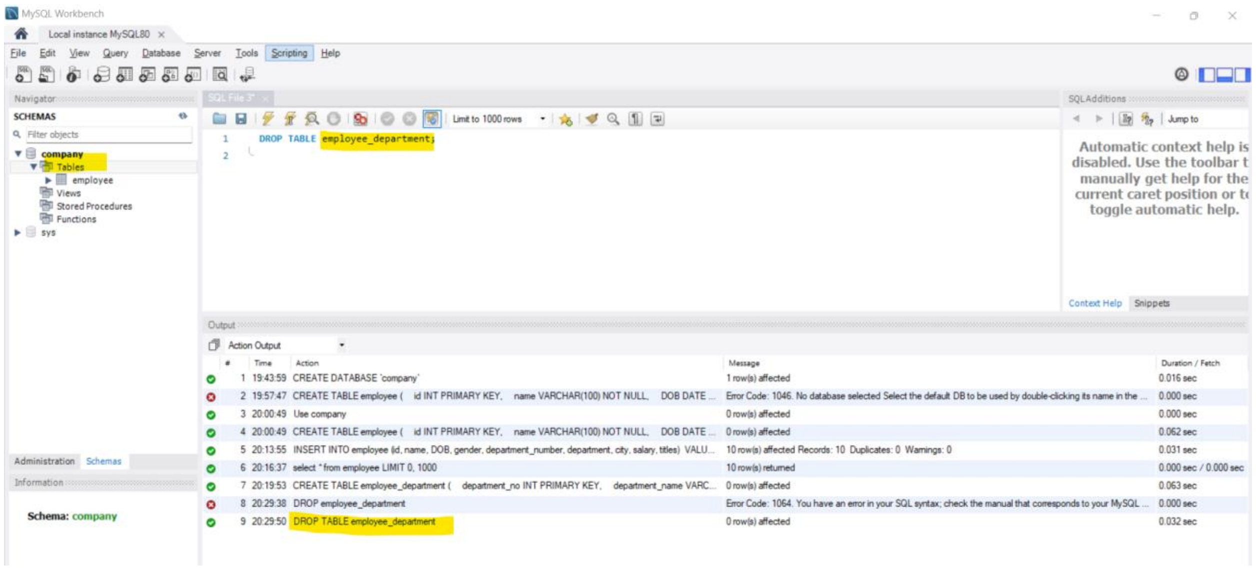
- **salary** (DECIMAL(10, 2)):
  This type stores money values accurately, with two decimal places. It's perfect for handling financial calculations precisely.
- **NOT NULL**
  It is used to make sure a field has a value and isn't left empty. This helps maintain data integrity by preventing missing or incomplete entries. It is mainly used for ‘ID’ values as every employee in company is assigned a unique ID and a department.

employee_department

- **department_no** (INT):
  This type is efficient for numeric identifiers and indexing. It ensures quick lookups and easy sorting.
- **department_name** (VARCHAR(100)):
  Varchar(100) handles variable-length data up to 100 characters. Usually, the data stored in these columns won’t exceed 100 characters.

5. Drop table employe

SQL:- DROP TABLE employee_department;



```
DROP TABLE employee_department;
```

#	Time	Action	Message	Duration / Fetch
1	19:43:59	CREATE DATABASE 'company'	1 row(s) affected	0.016 sec
2	19:57:47	CREATE TABLE employee (id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, DOB DATE ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the ...	0.000 sec
3	20:00:49	Use company	0 row(s) affected	0.000 sec
4	20:00:49	CREATE TABLE employee (id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, DOB DATE ...	0 row(s) affected	0.062 sec
5	20:13:55	INSERT INTO employee (id, name, DOB, gender, department_number, department, city, salary, titles) VALU...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
6	20:16:37	select * from employee LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
7	20:19:53	CREATE TABLE employee_department (department_no INT PRIMARY KEY, department_name VARC...	0 row(s) affected	0.063 sec
8	20:29:38	DROP employee_department	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ...	0.000 sec
9	20:29:50	DROP TABLE employee_department	0 row(s) affected	0.032 sec

6. **Which one's better approach?**
   -> Data Deletion by dropping the relation or Data Archival by renaming the relation. Provide your justification for each of the above specified approaches?

In my opinion, combining both approaches can be a great way to manage your data effectively. You can drop tables that you're sure you won't need any more to keep your database clean and fast. For data that might be useful later or needs to be kept for legal reasons, you can rename and archive it.

Both approaches to managing data have their own advantages and disadvantages, and the best decisions depend on the specific requirements and context of your system.

1. Data Deletion by Dropping the Relation

<u>Advantages:</u>

- Simplicity: Dropping a relation (table) is a straightforward operation. It completely removes the data and schema associated with the table, freeing up storage and simplifying database management.
- Performance: Removing unnecessary tables can improve database performance by reducing the size and complexity of the database, potentially speeding up queries and maintenance operations.
- Security: Completely removing data can be a way to ensure that sensitive information is no longer accessible, especially if the data is no longer needed and there are no compliance requirements to keep it.

<u>Disadvantages:</u>

- Irreversibility: Dropping a table is a permanent action. If you need the data in the future, it will be lost unless you have a backup. This can be problematic if the data might be needed for historical analysis or auditing.
- Backup and Recovery Complexity: If you frequently drop and recreate tables, keeping backups and making sure everything stays consistent can get complicated. Restoring data from backups can be slow and difficult.

2. Data Archival by Renaming the Relation

<u>Advantages:</u>

- Data Retention: Archiving by renaming allows you to retain the data for future reference or historical analysis. It ensures that you don't lose valuable information that might be needed later.
- Compliance: For many industries, data retention is required by legal or regulatory standards. Archiving by renaming helps meet these requirements by keeping data accessible but not active.
- Easy Retrieval: Archived data can be restored or queried if needed. This approach makes it easier to access historical data without needing to reconstruct it from backups.

<u>Disadvantages:</u>

- Database Bloat: Keeping archived tables in the same database can lead to increased size and complexity, which may impact performance. The database schema can become cluttered with old tables.

- Management Overhead: Maintaining and managing archived data adds overhead. You need to ensure that archived tables are properly managed and not accidentally queried or modified.
- Potential for Confusion: Renaming tables might lead to confusion if not well-documented. Users or administrators might not be aware of the purpose of the renamed tables, leading to potential mistakes or misunderstandings.

Justification for Each Approach

- **Drop Tables** when you're sure you don't need the data anymore, there are no legal requirements to keep it, and you want to keep your database fast and tidy. This is good for data that's old or not useful anymore.
- **Rename Tables for Archiving** when you need to keep old data for future use or legal reasons. This helps if the data might be useful later or if you must follow regulations about data retention.

In my previous organizations we had used combination of these approaches depending on the type of data and its importance. For example, active been managed in the primary database, while historical or less frequently accessed data archived separately

Task 2 - Higher Order Thinking Skills (HOTS)

7. Describe the potential ethical and privacy considerations when working with employee data in a database.

When working with employee data in a database, there are several important ethical and privacy considerations to keep in mind:

1. **Confidentiality:** You need to protect employee data from unauthorized access. Only people who need to see the data should be able to access it.
2. **Consent:** Employees should be aware of and agree to how their data is being used. They should know what data is collected and why.
3. **Data Accuracy:** Ensure the data is correct and updated. Mistakes in data can lead to incorrect decisions or unfair treatment.
4. **Data Minimization:** Only collect and keep the data that's necessary for your purposes. Avoid collecting extra information that isn't needed.
5. **Security:** Use strong security measures to protect the data from breaches or unauthorized access. This includes encryption and secure access controls.
6. **Transparency:** Be open about data collection practices and how the data is used. Employees should know their rights regarding their data.

When a data privacy breach happens, it can be very costly. **Hard costs** include fines from regulators and money spent on legal fees or compensating people affected by the breach. **Soft costs** involve damage to your company's reputation and a loss of trust from clients, which can hurt business and make it harder to keep employees motivated. Overall, protecting data is crucial to avoid these expensive and damaging consequences.

How would you ensure that the database complies with data protection regulations like GDPR or HIPAA?

HIPAA is focused on healthcare organizations and how personal health information is used in the US. GDPR, on the other hand, is a broader legislation that supervises any organization handling personally identifiable information of an EU or UK citizen.

To ensure compliance with data protection regulations like GDPR (in Europe) or HIPAA (in the U.S.), you can:

1. **Understand the Regulations:** Familiarize yourself with the rules of GDPR or HIPAA, depending on your location and the nature of the data.
2. **Obtain Consent:** Get clear permission from employees before collecting or using their data. Ensure they understand what they agree to.
3. **Implement Security Measures:** Use encryption, secure access controls, and other protections to keep data safe.
4. **Provide Access Rights:** Allow employees to access their own data and request corrections if needed.
5. **Limit Data Usage:** Only use data for the purposes you've stated and avoid sharing it unnecessarily.
6. **Regular Audits:** Periodically review your data practices and security measures to ensure they remain compliant with regulations.
7. **Encryption**: Encrypt data both in transit (while it's being sent) and at rest (while it's stored) to protect it from unauthorized access.
8. **Backup and Recovery**: Regularly back up data and have a recovery plan in place to protect against data loss or corruption.

By focusing on these considerations and steps, you can help protect employee data and comply with relevant regulations.

References

OneTrust. (n.d.). *HIPAA vs. GDPR compliance*. OneTrust. Retrieved, from **https://www.onetrust.com/blog/hipaa-vs-gdpr-compliance/**

ISACA. (2016). *An ethical approach to data privacy protection*. ISACA Journal, 6. Retrieved from **https://www.isaca.org/resources/isaca-journal/issues/2016/volume-6/an-ethical-approach-to-data-privacy-protection**

W3Schools. (n.d.). *SQL Tutorial*. Retrieved from **https://www.w3schools.com/sql/**