TO START:

- Create a new database named "CompanyDB."

```
CREATE DATABASE CompanyDB;
```

- Create a schema named "Sales" within the "CompanyDB" database.

```
CREATE SCHEMA Sales;
```

- Create a table named "employees" with columns: employee_id (INT), first_name (VARCHAR), last_name (VARCHAR), and salary (DECIMAL) within the "Sales" schema.

  Note: on employee_id use sequence **NOT** Identity.

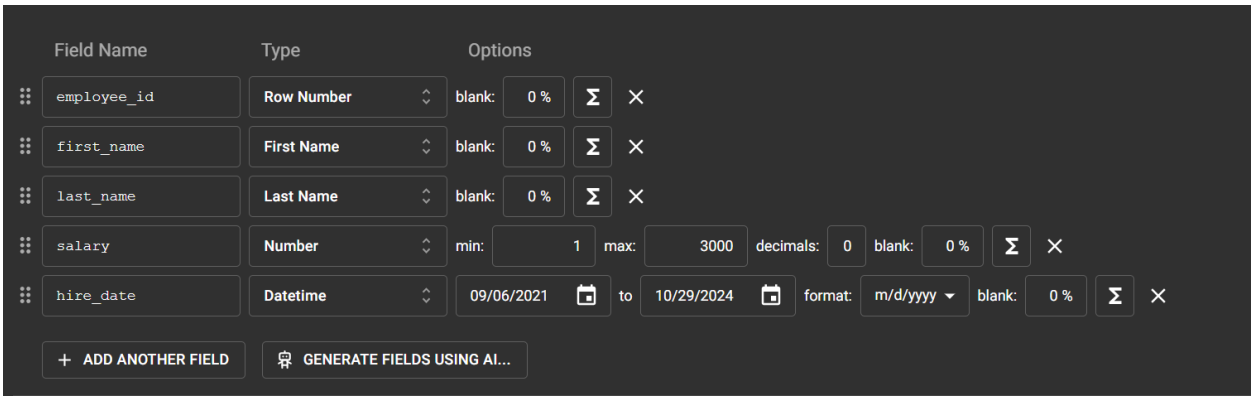
```
CREATE SEQUENCE Sales.cnt
start with 1
INCREMENT by 1;

CREATE TABLE Sales.employees
(
employee_id  INT PRIMARY KEY
DEFAULT (NEXT VALUE FOR Sales.cnt),
first_name  VARCHAR(30),
last_name  VARCHAR(30),
salary DECIMAL(10,2),
);
```

- Alter the "employees" table to add a new column named "hire_date" with the data type DATE.

```
ALTER TABLE Sales.employees
Add hire_date date;
```

- **Required:** Add mock data to this table use https://www.mockaroo.com



DATA MANIPULATION Exercises:

- Select all columns from the "employees" table.

```
SELECT *
FROM Sales.employees;
```

- Retrieve only the "first_name" and "last_name" columns from the "employees" table.

```
SELECT first_name,last_name
FROM Sales.employees;
```

- Retrieve “full name” as a one column from "first_name" and "last_name" columns from the "employees" table.

```
SELECT first_name+' '+last_name AS 'Full
Name'
FROM Sales.employees;
```

- Show the average salary of all employees.

```
SELECT AVG(Sales.employees.salary)  AS
'average of salary'
FROM Sales.employees;
```

- Select employees whose salary is greater than 50000.

```
SELECT *
FROM Sales.employees
WHERE Sales.employees.salary>5000;
```

- Retrieve employees hired in the year 2020.

```
SELECT *
FROM Sales.employees
WHERE year(hire_date) = '2020';
```

- List employees whose last names start with 'S.'

```
SELECT *
FROM Sales.employees
WHERE Sales.employees.last_name like 'S%';
```

- Display the top 10 highest-paid employees.

```
SELECT top(10) *
FROM Sales.employees
order by salary desc
```

- Find employees with salaries between 40000 and 60000.

```
SELECT *
FROM Sales.employees
WHERE salary between 40000 and 60000;
```

- Show employees with names containing the substring 'man.'

```
SELECT *
FROM Sales.employees
WHERE first_name like '%man%';
```

- Display employees with a NULL value in the "hire_date" column.

```
SELECT *
FROM Sales.employees
WHERE hire_date is null;
```

- Select employees with a salary in the set (40000, 45000, 50000).

```
SELECT *
FROM Sales.employees
WHERE salary in (40000 ,45000, 50000);
```

- Retrieve employees hired between '2020-01-01' and '2021-01-01.'

```
SELECT *
FROM Sales.employees
WHERE hire_date between '2020-01-01' and
'2021-01-01';
```

- List employees with salaries in descending order.

```
SELECT *
FROM Sales.employees
order by salary desc
```

- Show the first 5 employees ordered by "last_name" in ascending order.

```
SELECT top(5) *
FROM Sales.employees
order by last_name asc;
```

- Display employees with a salary greater than 55000 and hired in 2020.

```
SELECT *
FROM Sales.employees
```

```
WHERE salary >55000 and year(hire_date) =
'2020';
```

- Select employees whose first name is 'John' or 'Jane.'

```
SELECT *
FROM Sales.employees
WHERE first_name in ('John','Jane');
```

- List employees with a salary less than or equal to 55000 and a hire date after '2022-01-01.'

```
SELECT *
FROM Sales.employees
WHERE salary <= 55000 and hire_date >'2022-
01-01';
```

- Retrieve employees with a salary greater than the average salary.

```
SELECT *
FROM Sales.employees
WHERE salary > (select avg(salary) from
Sales.employees);
```

- Display the 3rd to 7th highest-paid employees.

```
SELECT *
FROM(
SELECT * ,ROW_NUMBER() OVER(order by salary
Desc) AS HS
FROM  Sales.employees
) as newTable
WHERE HS between 3 and 7;
```

- List employees hired after '2021-01-01' in alphabetical order.

```
SELECT *
FROM  Sales.employees
WHERE hire_date >'2021-01-01'
order by first_name,last_name;
```

- Retrieve employees with a salary greater than 50000 and last name not starting with 'A.'

```
select *
from Sales.employees
where salary >50000 and last_name not like
'A%'
```

- Display employees with a salary that is not NULL.

```
select *
from Sales.employees
where salary is not null;
```

- Show employees with names containing 'e' or 'i' and a salary greater than 45000.

```
select *
from Sales.employees
where salary >45000 and (first_name +' '+
last_name)  like '%e%' or
(first_name +' '+ last_name)  like '%i%';
```