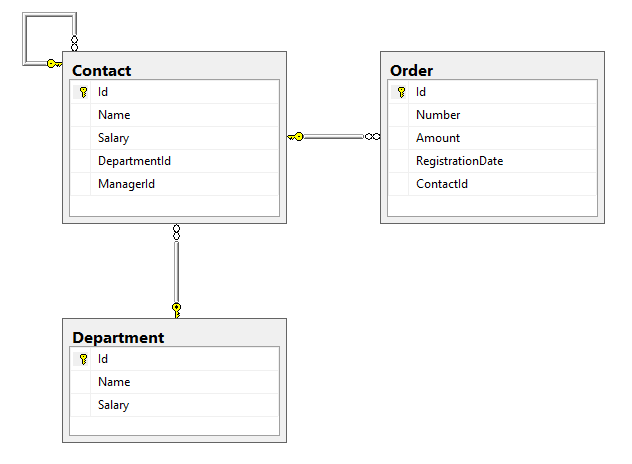
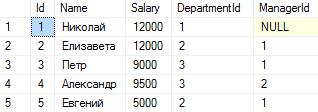
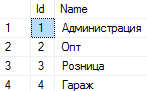
**SQL**

1. **Дана структура данных:** 

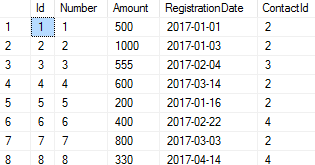
**Contact**



**Department**



**Order**



**2.а. Напишите запрос который покажет среднюю сумму заказа каждого сотрудника в разрезе месяцев.**

|  |
| --- |
| *SELECT DISTINCT cont.Name, MONTH(ord.RegistrationDate) [Month], (SELECT AVG(Amount) FROM [Order] WHERE MONTH(RegistrationDate) = MONTH(ord.RegistrationDate)) [Average]*  *FROM DBO.Contact cont*  *LEFT JOIN DBO.[Order] ord*  *ON cont.Id = ord.ContactId*  *GROUP BY MONTH(ord.RegistrationDate), ord.RegistrationDate, cont.Name ORDER BY [Month]* |

**2.b. Напишите запрос, который удалит всех сотрудников, у которых нет заказов. Сотрудников может быть n миллионов. Как избежать проблемы с производительностью?**

|  |
| --- |
| *WHILE @@ROWCOUNT <> 0*  *DELETE TOP (5) Contact WHERE Id NOT IN (SELECT ContactId FROM [Order])* |

**2.c. Какие индексы и ограничения нужно добавить в структуру таблиц? Почему?**

|  |
| --- |
| Обычные индексы на внешние ключи для ускорения join и фильтрации, остальные – в зависимости от предполагаемого использования таблицы. Если будет производиться много вставок/изменений, количество индексов должно быть минимальным. Если предполагается преимущественно выборки, тогда имеет смысл создавать индексы на поля, по которым будет чаще всего производится выборка.   Вставить ограничение NOT NULL на все обязательные поля, остальные – зависит от внешних требований. |

*P(R)* represents a pattern drawn by Julia in *R* rows. The following pattern represents *P(5)*:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Write a query to print the pattern *P(20)*.

|  |
| --- |
| DECLARE @total INT = 20, @current INT = 1, @str varchar(100), @tmp INT = 0;  WHILE (@current <= @total)  BEGIN  WHILE (@tmp < @current)  BEGIN  SET @str = CONCAT(@str, '\*');  SET @tmp = @tmp + 1;  END  PRINT(@str);  SET @current = @current + 1;  END |