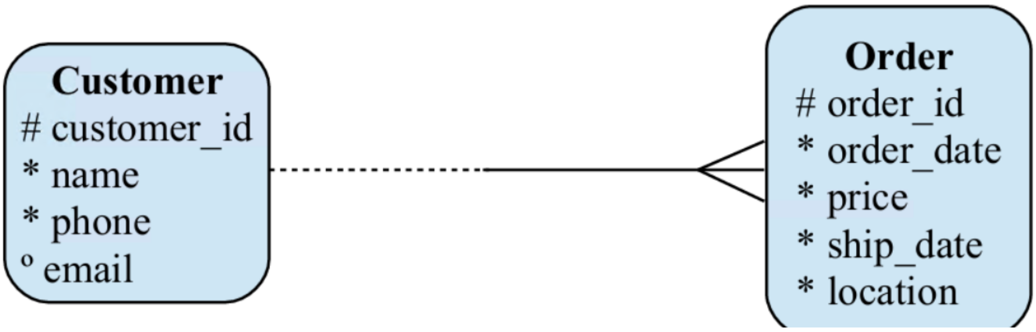
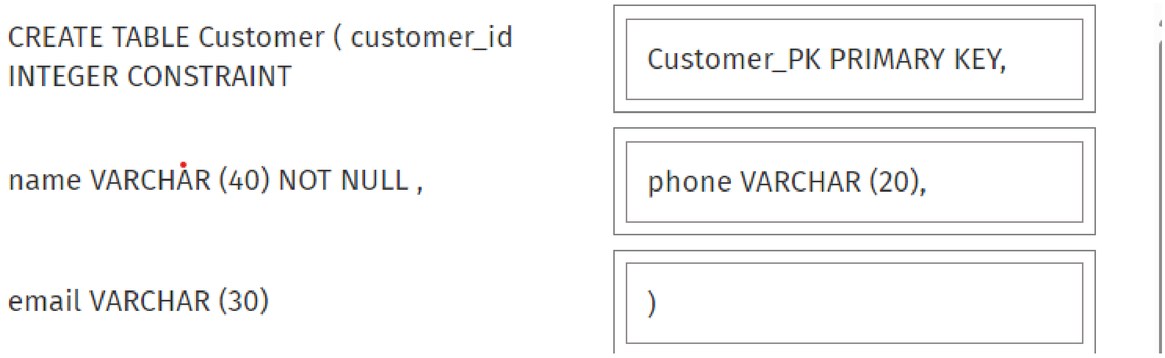
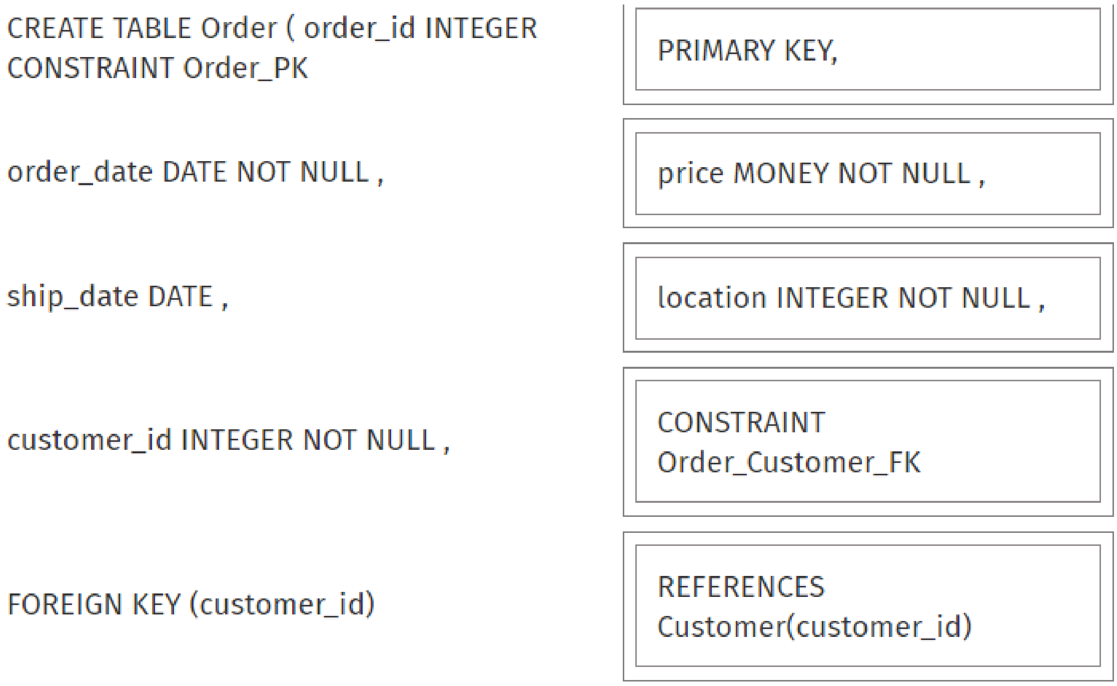
1

Consider a binary one-to-many relationship (A customer can place many orders. An order is always assigned to exactly one customer.). Prepare the statements in the correct order to implement tables for the following diagram.



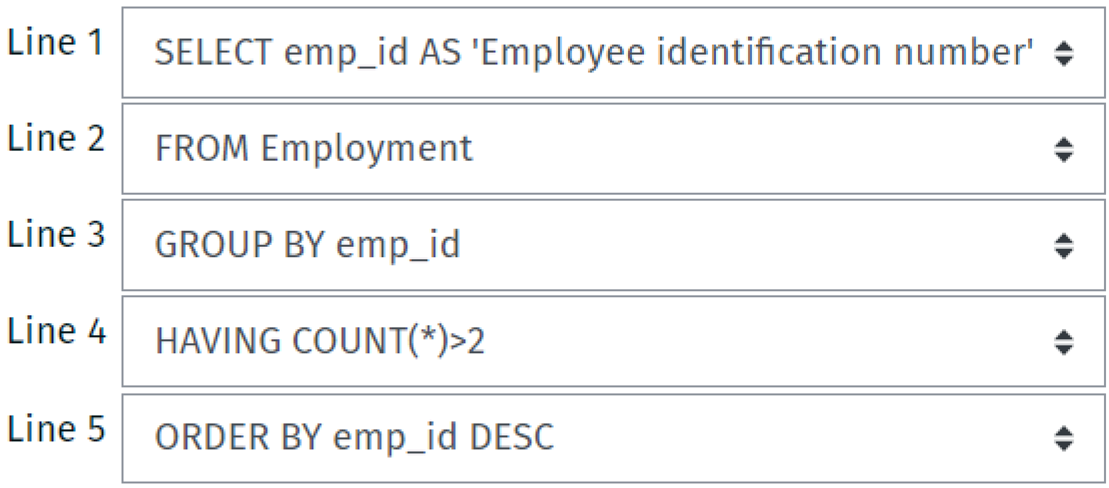


CREATE TABLE Customer (customer_id INTEGER CONSTRAINT	Customer_PK PRIMARY KEY,
name VARCHAR (40) NOT NULL ,	phone VARCHAR (20),
email VARCHAR (30)	)



CREATE TABLE Order (order_id INTEGER CONSTRAINT Order_PK	PRIMARY KEY,
order_date DATE NOT NULL ,	price MONEY NOT NULL ,
ship_date DATE ,	location INTEGER NOT NULL ,
customer_id INTEGER NOT NULL ,	CONSTRAINT Order_Customer_FK
FOREIGN KEY (customer_id)	REFERENCES Customer(customer_id)

2. Determine employee identification number if he/she has more than two rows in the Employment table. Arrange the result based on identification numbers in descending order.



Line 1 SELECT emp_id AS 'Employee identification number'

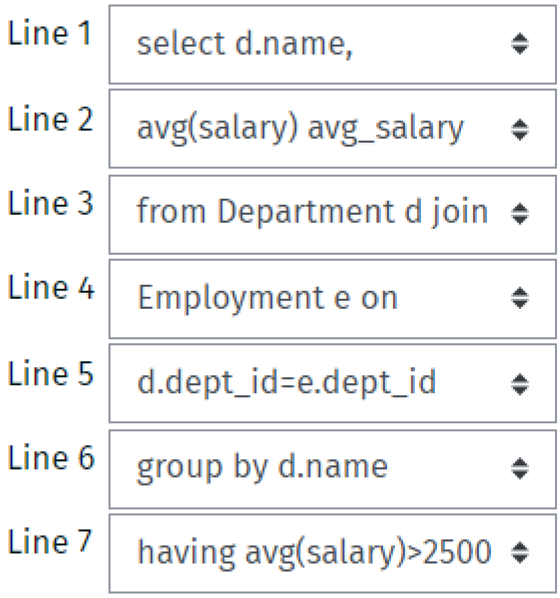
Line 2 FROM Employment

Line 3 GROUP BY emp_id

Line 4 HAVING COUNT(*)>2

Line 5 ORDER BY emp_id DESC

3. Select a department name and its average salary if the average salary in this department is greater than 2500.



Line 1 select d.name,

Line 2 avg(salary) avg_salary

Line 3 from Department d join

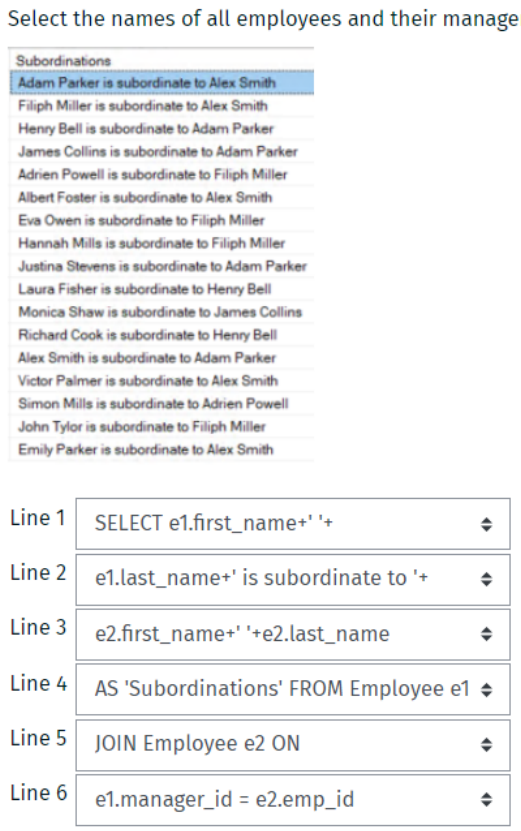
Line 4 Employment e on

Line 5 d.dept_id=e.dept_id

Line 6 group by d.name

Line 7 having avg(salary)>2500

4.



Select the names of all employees and their managers in the following form

Subordinations
Adam Parker is subordinate to Alex Smith
Filiph Miller is subordinate to Alex Smith
Henry Bell is subordinate to Adam Parker
James Collins is subordinate to Adam Parker
Adrien Powell is subordinate to Filiph Miller
Albert Foster is subordinate to Alex Smith
Eva Owen is subordinate to Filiph Miller
Hannah Mills is subordinate to Filiph Miller
Justina Stevens is subordinate to Adam Parker
Laura Fisher is subordinate to Henry Bell
Monica Shaw is subordinate to James Collins
Richard Cook is subordinate to Henry Bell
Alex Smith is subordinate to Adam Parker
Victor Palmer is subordinate to Alex Smith
Simon Mills is subordinate to Adrien Powell
John Tylor is subordinate to Filiph Miller
Emily Parker is subordinate to Alex Smith

Line 1 SELECT e1.first_name+' '+

Line 2 e1.last_name+' is subordinate to '+

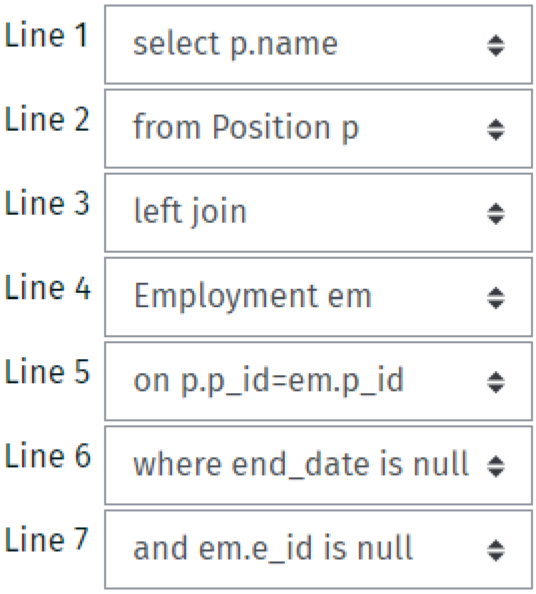
Line 3 e2.first_name+' '+e2.last_name

Line 4 AS 'Subordinations' FROM Employee e1

Line 5 JOIN Employee e2 ON

Line 6 e1.manager_id = e2.emp_id

5. Select the names of positions which are currently not occupied by current employees.



Line 1 select p.name

Line 2 from Position p

Line 3 left join

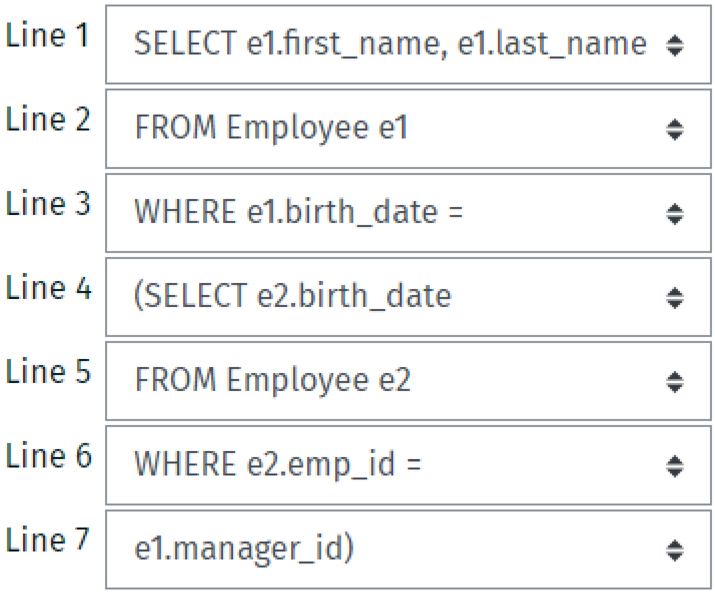
Line 4 Employment em

Line 5 on p.p_id=em.p_id

Line 6 where end_date is null

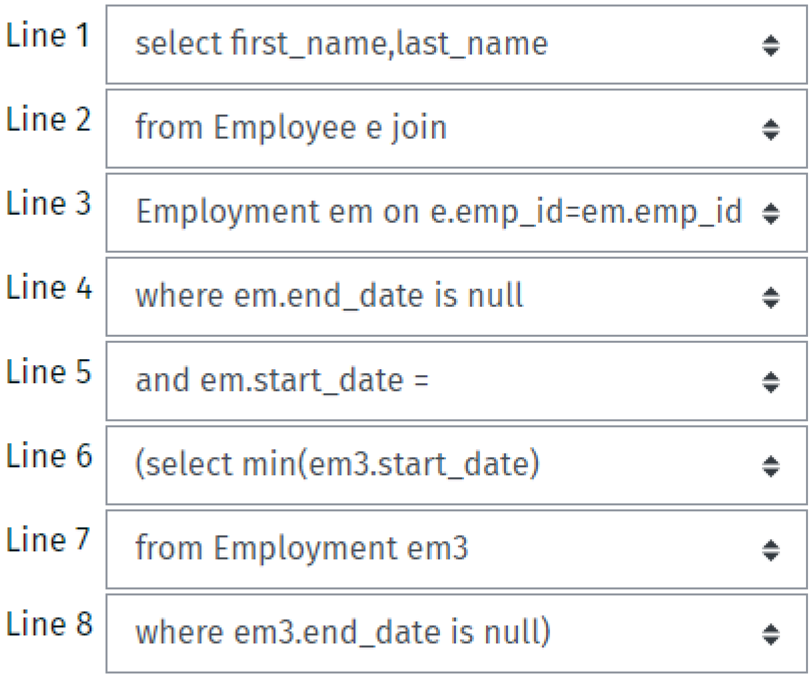
Line 7 and em.e_id is null

6 Get employees (first and last names) which are the age of their managers.



Line 1	SELECT e1.first_name, e1.last_name
Line 2	FROM Employee e1
Line 3	WHERE e1.birth_date =
Line 4	(SELECT e2.birth_date
Line 5	FROM Employee e2
Line 6	WHERE e2.emp_id =
Line 7	e1.manager_id)

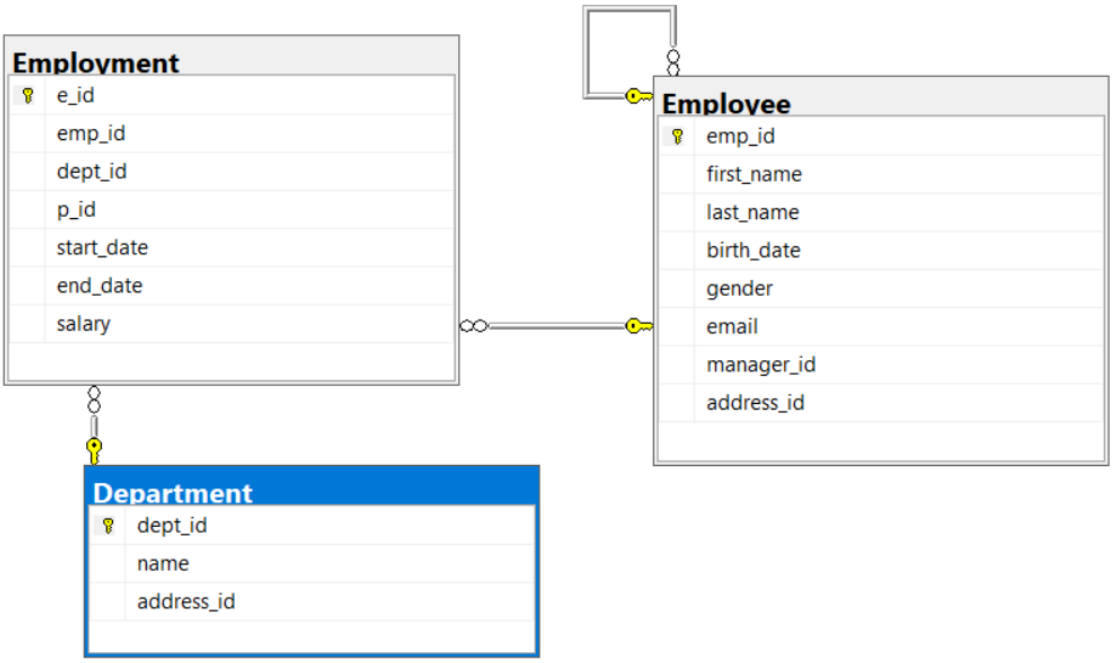
7. Print the names of each current employee who is employed the longest.



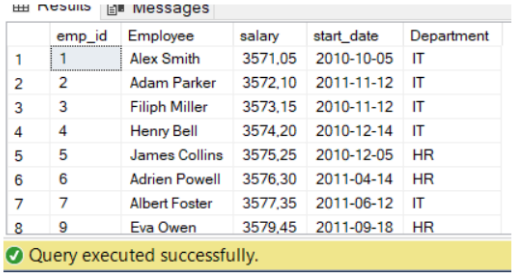
Line 1	select first_name,last_name
Line 2	from Employee e join
Line 3	Employment em on e.emp_id=em.emp_id
Line 4	where em.end_date is null
Line 5	and em.start_date =
Line 6	(select min(em3.start_date)
Line 7	from Employment em3
Line 8	where em3.end_date is null)

8

Consider the following diagram

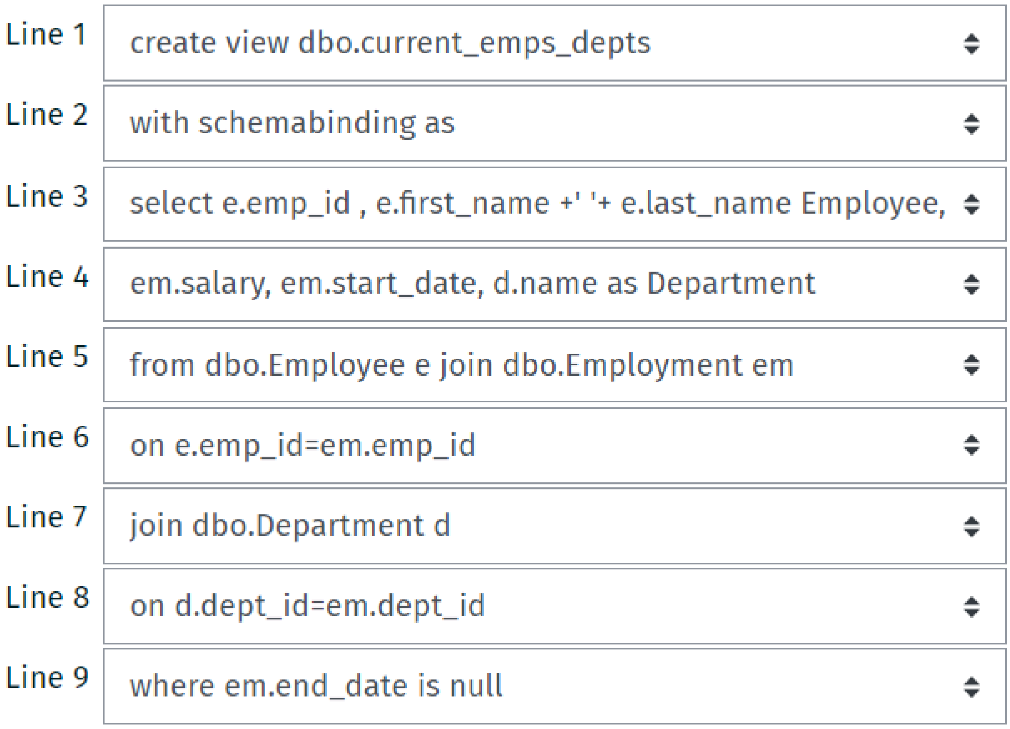


and implement the view called dbo.current_emps_depts which involves the following data for current employees (the base tables cannot be modified in a way that would affect the view definition)



	emp_id	Employee	salary	start_date	Department
1	1	Alex Smith	3571,05	2010-10-05	IT
2	2	Adam Parker	3572,10	2011-11-12	IT
3	3	Filiph Miller	3573,15	2010-11-12	IT
4	4	Henry Bell	3574,20	2010-12-14	IT
5	5	James Collins	3575,25	2010-12-05	HR
6	6	Adrien Powell	3576,30	2011-04-14	HR
7	7	Albert Foster	3577,35	2011-06-12	IT
8	9	Eva Owen	3579,45	2011-09-18	HR

✓ Query executed successfully.



Line 1 create view dbo.current_emps_depts ⬍

Line 2 with schemabinding as ⬍

Line 3 select e.emp_id , e.first_name +' '+ e.last_name Employee, ⬍

Line 4 em.salary, em.start_date, d.name as Department ⬍

Line 5 from dbo.Employee e join dbo.Employment em ⬍

Line 6 on e.emp_id=em.emp_id ⬍

Line 7 join dbo.Department d ⬍

Line 8 on d.dept_id=em.dept_id ⬍

Line 9 where em.end_date is null ⬍

9.

The following statements have been run correctly.

```
CREATE TABLE dbo.Student ( [index] varchar(10) CONSTRAINT PK_Student_Index PRIMARY KEY,
first_name varchar(25) NOT NULL,
last_name varchar(30) NOT NULL,
birth_date date NOT NULL,
gender char(1) CONSTRAINT CH_Employee_gen CHECK(gender='M' OR gender='F'),
email varchar(40) )
go
```

```
create view dbo.view_student
with schemabinding
as
select [index], first_name, last_name, email, birth_date
from dbo.Student
where datediff(year, birth_date,getdate())>18
go
```

What of the following statements fail?

☑ a. alter view dbo.view_student add column gender char(1)

☑ b. alter table dbo.student drop column email

☑ c. alter table dbo.student add column faculty_id int

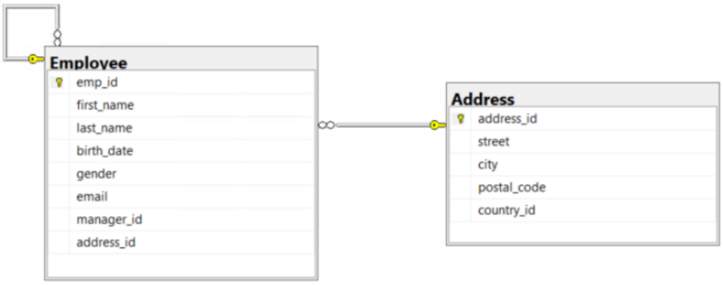
□ d. alter table dbo.student drop constraint PK_Student_Index

☑ e. alter table dbo.student drop column gender

10

Create the table Employee for the following diagram. The manager_id column is a foreign key which points to emp_id. The address_id column in Employee is a foreign key into the Address table. The email column should prevent inserting duplicates. Only two values are allowed to be inserted into the gender column : M and F.

The table Address exists.



Line 1 CREATE TABLE Employee(⬍

Line 2 emp_id int IDENTITY(1,1) CONSTRAINT PK_Employee_Emp_id PRIMARY KEY, ⬍

Line 3 first_name varchar(25) CONSTRAINT NN_Employee_fn NOT NULL, last_name varchar(30) NOT NULL, ⬍

Line 4 birth_date date NOT NULL, gender char(1) CONSTRAINT CH_Employee_gen CHECK(gender='M' OR gender='F'), ⬍

Line 5 email varchar(40) CONSTRAINT U_Employee_Email UNIQUE, ⬍

Line 6 manager_id int, ⬍

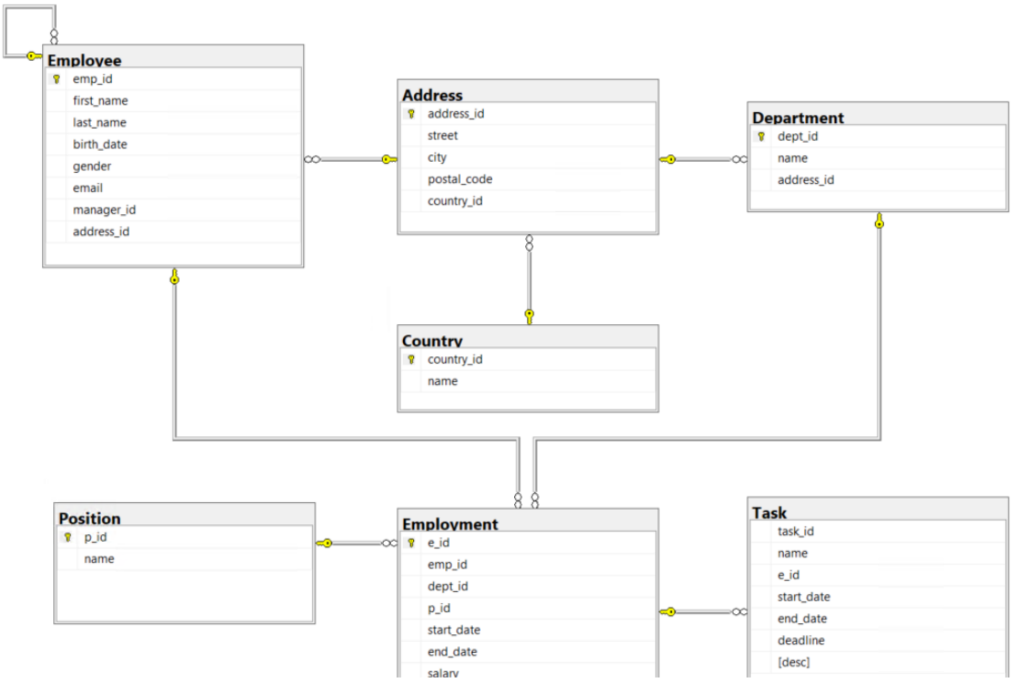
Line 7 address_id int foreign key references Address(address_id), ⬍

Line 8 CONSTRAINT FK_Employee_Mgr_id2 FOREIGN KEY(manager_id) ⬍

Line 9 REFERENCES Employee(emp_id)) ⬍

11

Which of the following statements will fail against the below database schema? It is assumed that each statement is run independently.

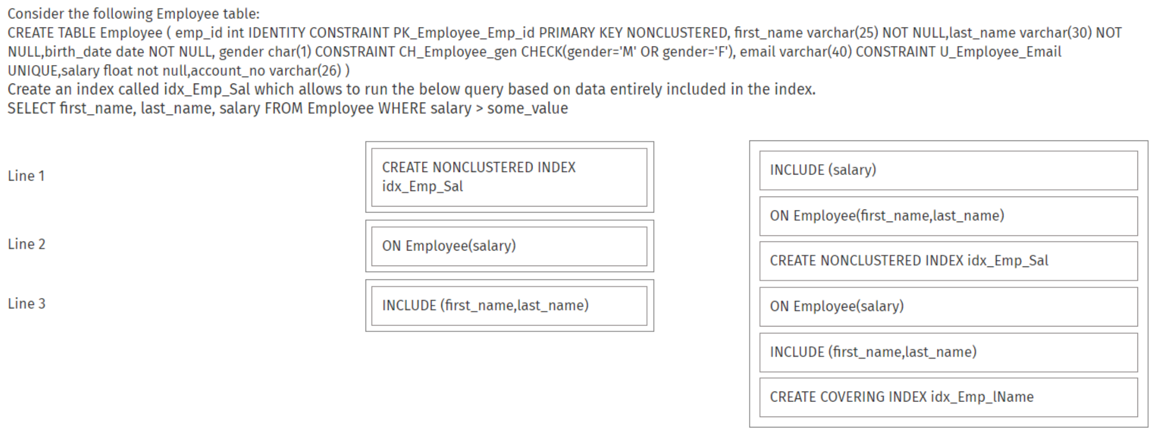


☑ a. drop table Country

☑ b. drop table Employee

☑ c. drop table Employment

☐ d. drop table task



Consider the following Employee table:
CREATE TABLE Employee (emp_id int IDENTITY CONSTRAINT PK_Employee_Emp_id PRIMARY KEY NONCLUSTERED, first_name varchar(25) NOT NULL,last_name varchar(30) NOT NULL,birth_date date NOT NULL, gender char(1) CONSTRAINT CH_Employee_gen CHECK(gender='M' OR gender='F'), email varchar(40) CONSTRAINT U_Employee_Email UNIQUE,salary float not null,account_no varchar(26))
Create an index called idx_Emp_Sal which allows to run the below query based on data entirely included in the index.
SELECT first_name, last_name, salary FROM Employee WHERE salary > some_value

Line 1	CREATE NONCLUSTERED INDEX idx_Emp_Sal
Line 2	ON Employee(salary)
Line 3	INCLUDE (first_name,last_name)

INCLUDE (salary)

ON Employee(first_name,last_name)

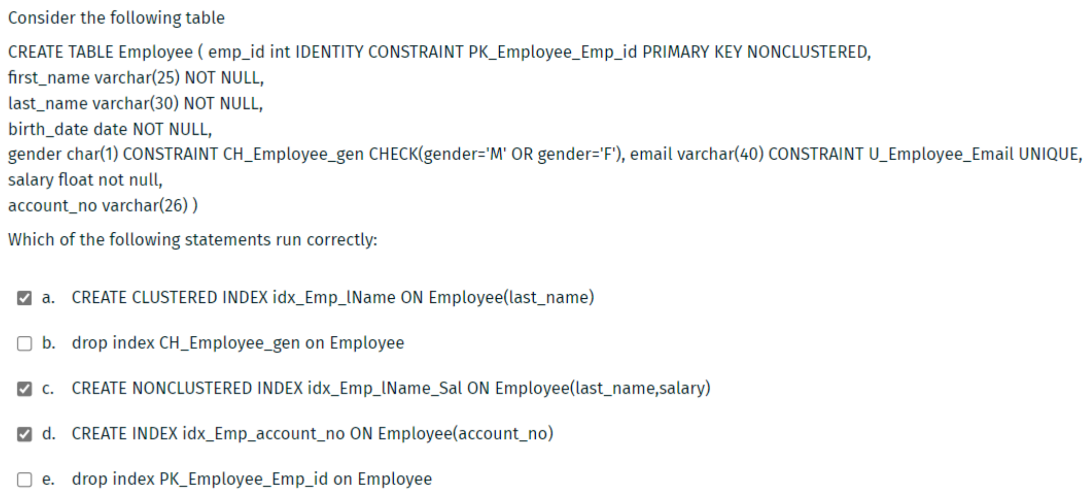
CREATE NONCLUSTERED INDEX idx_Emp_Sal

ON Employee(salary)

INCLUDE (first_name,last_name)

CREATE COVERING INDEX idx_Emp_lName

13



Consider the following table

CREATE TABLE Employee (emp_id int IDENTITY CONSTRAINT PK_Employee_Emp_id PRIMARY KEY NONCLUSTERED,
first_name varchar(25) NOT NULL,
last_name varchar(30) NOT NULL,
birth_date date NOT NULL,
gender char(1) CONSTRAINT CH_Employee_gen CHECK(gender='M' OR gender='F'), email varchar(40) CONSTRAINT U_Employee_Email UNIQUE,
salary float not null,
account_no varchar(26))

Which of the following statements run correctly:

☑ a. CREATE CLUSTERED INDEX idx_Emp_lName ON Employee(last_name)

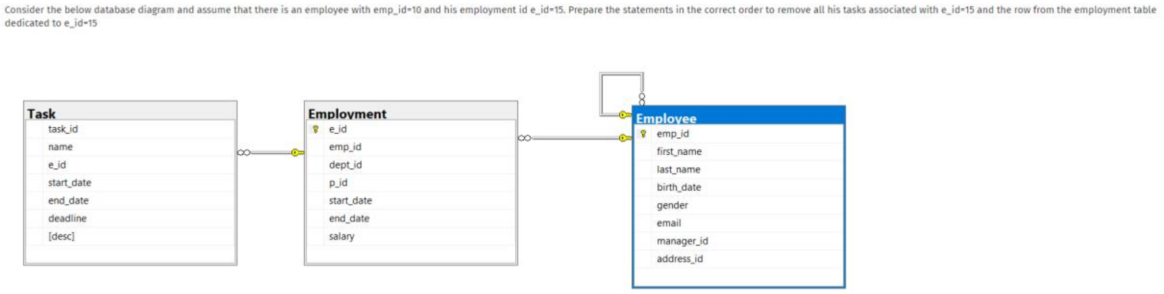
☐ b. drop index CH_Employee_gen on Employee

☑ c. CREATE NONCLUSTERED INDEX idx_Emp_lName_Sal ON Employee(last_name,salary)

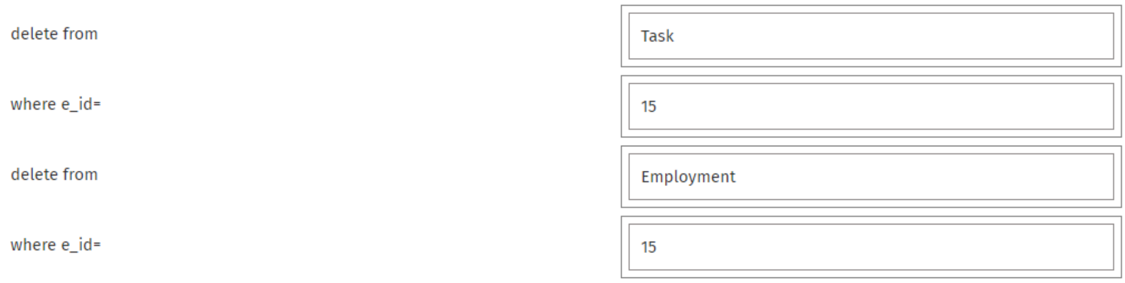
☑ d. CREATE INDEX idx_Emp_account_no ON Employee(account_no)

☐ e. drop index PK_Employee_Emp_id on Employee

14



Consider the below database diagram and assume that there is an employee with emp_id=10 and his employment id e_id=15. Prepare the statements in the correct order to remove all his tasks associated with e_id=15 and the row from the employment table dedicated to e_id=15



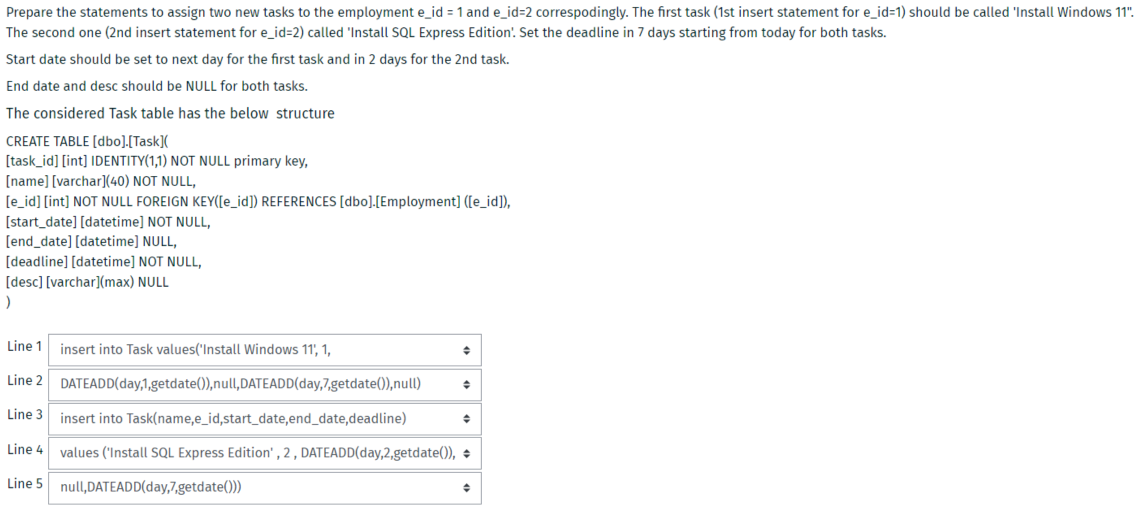
delete from Task

where e_id= 15

delete from Employment

where e_id= 15

15



Prepare the statements to assign two new tasks to the employment e_id = 1 and e_id=2 correspodingly. The first task (1st insert statement for e_id=1) should be called 'Install Windows 11". The second one (2nd insert statement for e_id=2) called 'Install SQL Express Edition'. Set the deadline in 7 days starting from today for both tasks.

Start date should be set to next day for the first task and in 2 days for the 2nd task.

End date and desc should be NULL for both tasks.

The considered Task table has the below structure

```
CREATE TABLE [dbo].[Task](
[task_id] [int] IDENTITY(1,1) NOT NULL primary key,
[name] [varchar](40) NOT NULL,
[e_id] [int] NOT NULL FOREIGN KEY([e_id]) REFERENCES [dbo].[Employment] ([e_id]),
[start_date] [datetime] NOT NULL,
[end_date] [datetime] NULL,
[deadline] [datetime] NOT NULL,
[desc] [varchar](max) NULL
)
```

Line 1 insert into Task values('Install Windows 11', 1,

Line 2 DATEADD(day,1,getdate()),null,DATEADD(day,7,getdate()),null)

Line 3 insert into Task(name,e_id,start_date,end_date,deadline)

Line 4 values ('Install SQL Express Edition' , 2 , DATEADD(day,2,getdate()),

Line 5 null,DATEADD(day,7,getdate()))