


Задача

В данной задаче необходимо уметь пользоваться хинтами по изменению плана запроса встроенного оптимизатора («взять управление на себя»), т.е. написать запрос с указанием конкретного индекса к каждой таблице, чтобы оптимизатор использовал указанные разработчиком индексы и очередность таблиц.

Существует временная таблица #A:

#A	
	DealID: numeric(15,0)
ParentCnt: int ChildCnt: int	

Данная таблица содержит уникальные идентификаторы сделок (DealID) на определённую дату. Поля ParentCnt и ChildCnt равны нулю.

Есть таблица tDealRelation:

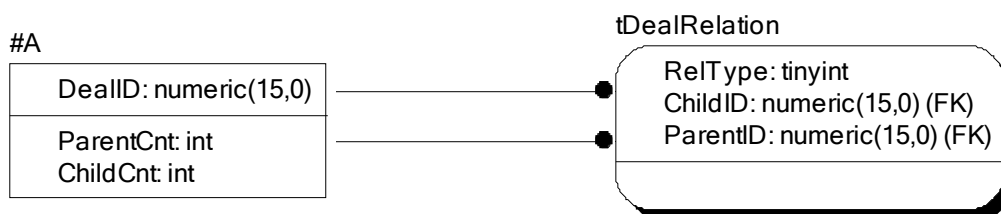
tDealRelation	
RelType: tinyint	
ParentID: numeric(15,0)	
ChildID: numeric(15,0)	

Таблица имеет следующие индексы

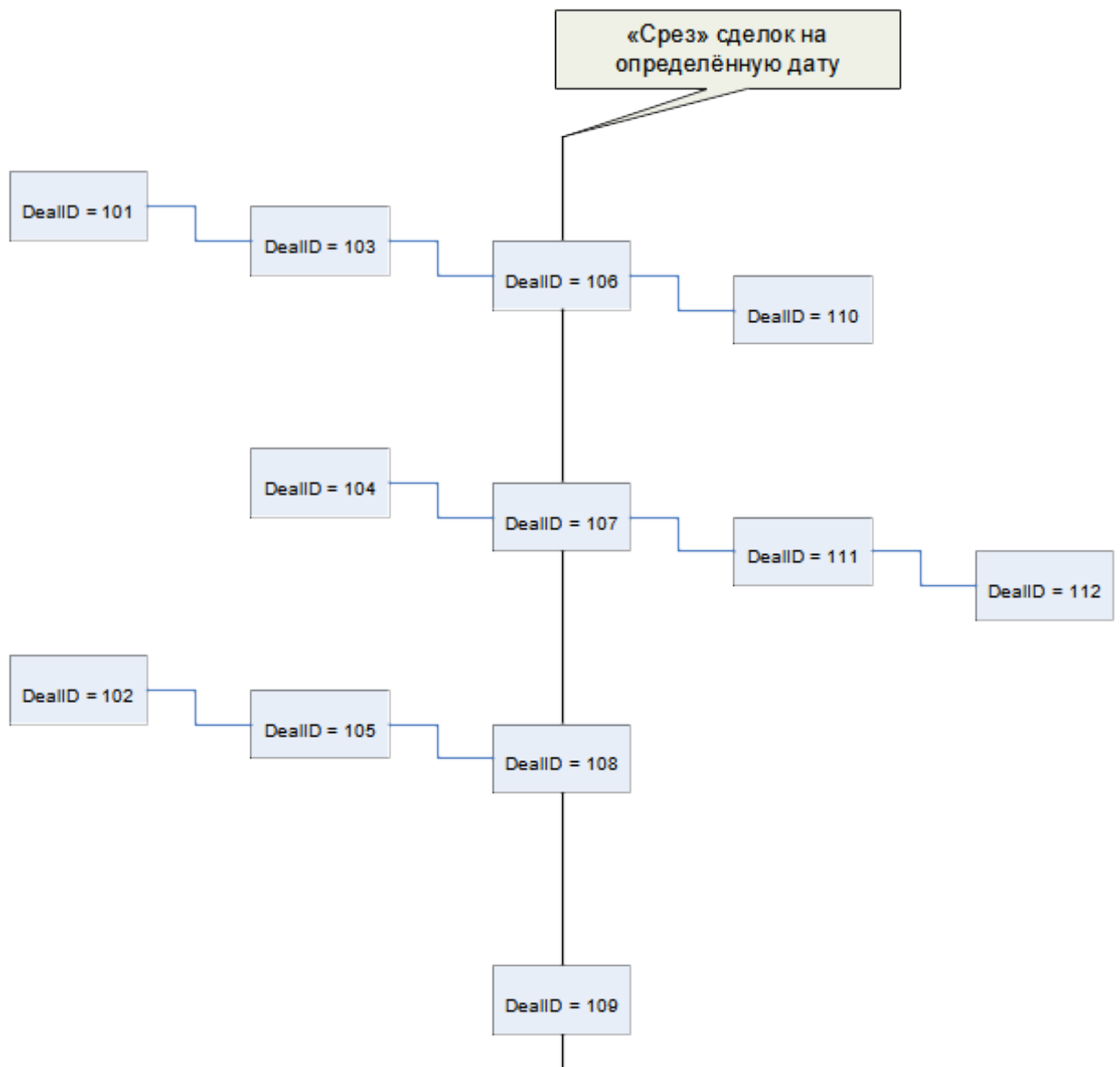
XIE2tDealRelation	RelType, ChildID	nonclustered
XIE3tDealRelation	ParentID, ChildID	nonclustered
XIE4tDealRelation	ChildID, RelType, ParentID	nonclustered
XPktDealRelation	RelType, ParentID, ChildID	nonclustered, unique

Данная таблица содержит записи, соответствующие связям типа «родитель-потомок» между сделками. Тип связи определён: RelType = 2. Цепочки сделок могут быть произвольной длины.

Связь таблиц выглядит следующим образом:



Ниже приведён пример цепочек сделок и их «отображения» в таблицах.



Заполнение таблиц для данного примера:

Таблица #А

DealID	ParentCnt	ChildCnt
106	0	0
107	0	0
108	0	0
109	0	0

Таблица tDealRelation

DealRelationID	RelType	ParentID	ChildID
1001	2	101	103
1002	2	102	105
1003	2	103	106
1004	2	104	107
1005	2	105	108
1006	2	106	110
1007	2	107	111
1008	2	111	112

Необходимо написать скрипт, который в таблице #А проставит в полях ParentCnt и ChildCnt количество потомков и предков для данного DealID.

Для приведённого выше примера таблица #A в результате выполнения скрипта выглядела бы следующим образом:

Таблица #A

DealID	ParentCnt	ChildCnt
106	2	1
107	1	2
108	2	0
109	0	0

При написании запросов необходимо использовать существующие индексы для таблицы tDealRelation.

Дополнительно: Оформить в функцию расчет количества предков для определенного DealID