

Первый этап

Индекс	Время выполнения
-	1330.00 мс
<pre>USE PaymentData GO -- Создание кластерного индекса на таблице Payment CREATE CLUSTERED INDEX iPayment_PayerPayeeDate ON dbo.Payment (Payer, Payee, Date) GO -- Создание некластерного индекса на таблице PaymentParticipant CREATE NONCLUSTERED INDEX iPaymentParticipant_Balance ON dbo.PaymentParticipant (Balance) INCLUDE (Oid, Name, ObjectType, ActiveFrom, InactiveFrom, BankDetails, Balance2, Balance3) GO -- Создание некластерного индекса на таблице Project CREATE NONCLUSTERED INDEX iProject_Client ON dbo.Project (Client) GO -- Создание некластерного индекса на таблице Project CREATE NONCLUSTERED INDEX iProject_Manager_Foreman ON dbo.Project (Manager, Foreman) GO -- Создание некластерного индекса на таблице PaymentCategory CREATE NONCLUSTERED INDEX iPaymentCategory_Flags ON dbo.PaymentCategory (ProfitByMaterial, CostByMaterial, NotInPaymentParticipantProfit) GO</pre>	1353.00 мс

(100 – операций)

Изначально мы решили сделать «комплексное обновление»: накинуть сразу много индексов, которые потенциально могли бы помочь производительности. Но, как показано в таблице, результаты не показали прироста (скорее наоборот). Мы подумали, что необходимо увеличить число итераций в тестировании (как минимум), и делать менее «комплексные» обновления, а добавлять индексы поочередно.

Второй этап

Индекс	Время выполнения
- (т.е. все дефолтно)	34103.00 мс
<pre>CREATE NONCLUSTERED INDEX iPayment_Date ON dbo.Payment (Date)</pre>	28363.00 мс
<pre>-- Создание некластерного индекса на столбце CostByMaterial CREATE NONCLUSTERED INDEX iPaymentCategory_CostByMaterial ON dbo.PaymentCategory (CostByMaterial) INCLUDE (Oid, Name, OptimisticLockField, GCRecord, ProfitByMaterial, NotInPaymentParticipantProfit) GO -- Создание некластерного индекса на столбце ProfitByMaterial CREATE NONCLUSTERED INDEX iPaymentCategory_ProfitByMaterial ON dbo.PaymentCategory (ProfitByMaterial) INCLUDE (Oid, Name, OptimisticLockField, GCRecord, CostByMaterial, NotInPaymentParticipantProfit) GO</pre>	26583.00 мс
<pre>CREATE NONCLUSTERED INDEX iPayment_Date ON dbo.Payment (Date) -- Создание некластерного индекса на столбце CostByMaterial CREATE NONCLUSTERED INDEX iPaymentCategory_CostByMaterial ON dbo.PaymentCategory (CostByMaterial) INCLUDE (Oid, Name, OptimisticLockField, GCRecord, ProfitByMaterial, NotInPaymentParticipantProfit) GO -- Создание некластерного индекса на столбце ProfitByMaterial CREATE NONCLUSTERED INDEX iPaymentCategory_ProfitByMaterial ON dbo.PaymentCategory (ProfitByMaterial)</pre>	25886.00 мс

<code>INCLUDE (Oid, Name, OptimisticLockField, GCRecord, CostByMaterial, NotInPaymentParticipantProfit) GO</code>	
---	--

(1000 операций)

Мы увеличили число итераций, добавили индекс на дату.

Идея: данные о платежах чаще всего добавляются в хронологическом порядке (хотя в теории может быть и не так), поэтому хранить их в отсортированном виде, при этом не перестраивая индекс с каждой операцией вставки кажется неплохой идеей

Затем мы проанализировали триггер T_payment_AI, решили посмотреть на индексы PaymentCategory, обнаружили там явную недостаточность 😊 Таблица используется на постоянной основе, т.к. платеж зависит от типа, следовательно добавление индексов в теории должно было дать прирост, в чем мы убедились на практике, добавив пару индексов

Затем мы объединили все три индекса вместе и получили итоговый результат