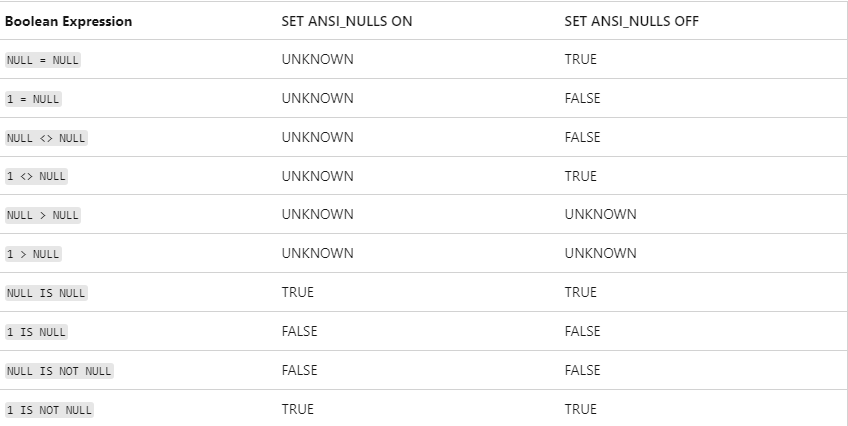
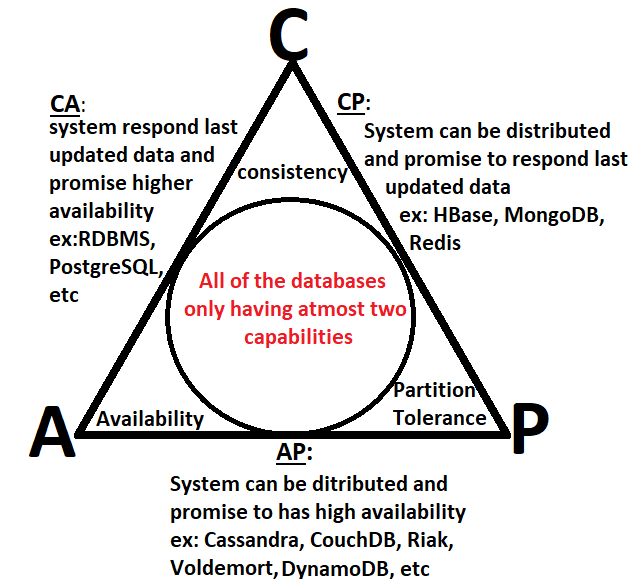
1. Transaction چیست و در چه جاهایی از آن باید استفاده کرد؟
   1. Transaction مثل یه all or none هست؛ یعنی یه سری کوئری یا دستور یا همه تا انتها به درستی اجرا میشن یا هیچ کدوم اجرا نمیشن.
2. قواعد ACID چیست؟
   1. در ادامه بحث transaction، برای اینکه حالت all or none تضمین شده باشه، یه سری قواعد هست که اگه رعایت بشه و بدونیم که توی transaction ما وجود دارن اونوقت میتونیم بگیم که all or none روی transaction اعمال شده. این قواعد به صورت زیر هستن:
      1. Atomicity: این یعنی یا کل transaction از اول تا آخر تمام و کمال اجرا بشه یا اصلا هیچ قسمتیش اجرا نشه. یه جورایی 90 درصد بحث all or none مال همین قاعده اس.
      2. Consistency: ما روی بیزینسی که داریم کار میکنیم، یه سری قانون یا ویژگی داریم که باید رعایت بشن. منظور از این قاعده، رعایت کردن اون قوانین یا ویژگی های بیزنسی هست و همچنین درست بودن دیتا بعد از انجام transaction. یه مثال میزنیم: فرض کنیم میخوایم کارت به کارت کنیم، مبلغی که میخوایم جابجا کنیم اگه بیشتر از موجودی ما باشه اونوقت موجودی حساب منفی میشه و این با قانون بیزنس بانکی در تضاده؛ پس تراکنش باید این قانون رو در زمان اجرا در نظر بگیره در غیر این صورت consistent نیست.
      3. Isolation: قاعده isolation میگه که اجرای چندین transaction در آن واحد نباید consistency رو نقض کنه. فرض کنیم در آن واحد چندتا transaction میخواد اجرا بشه و یکی میخواد دیتایی رو بخونه و اون یکی میخواد همون دیتا رو تغییر بده. اگه دقیقا زمانی که تغییر میخواد انجام بشه، دیتا توسط اون یکی transaction خونده بشه، اونوقت ما انگار دو مقدار متفاوت برای اون دیتا داریم و این consistency دیتا رو نقض میکنه. در واقع نباید زمانی که یه transaction داره یه دیتا رو تغییر میده (و هنوز اون تغییر انجام نشده) یه transaction دیگه بیاد و همون دیتا رو بخونه یا تغییر بده.
      4. Durability: قاعده آخر بحث نگهداری و موندگاری دیتاس؛ زمانی که یه transaction اجرا شه، هر اتفاقی هم که بیفته دیگه اثر اون روی دیتابیس پروژه موندگاره. حالا میخواد ارور خوردن سیستم باشه یا سخت افزار یا هر چیز دیگه ای؛ این موارد باعث پاک شدن اثر اون transaction نمیشه.
3. دستور SAVE POINT؟
   1. مثل یه COMMIT موقته؛ اگه جایی ROLLBACK داشته باشیم و بخوایم یه تیکه ای که قبلا اجرا شده شامل این ROLLBACK نشه، میایم SAVE POINT میذاریم که اون تیکه محفوظ بمونه اما نتیجه اش هنوز روی دیسک ذخیره نشده.
4. تفاوت datetime و datetime2 در چیست؟
   1. ویژگی های Datetime:
      1. 8 بایت حافظه اشغال می کنه
      2. در میزان دقت، دیتا رو به ضرایب 0.000, 0.003, 0.007 رند می کنه.
      3. شروع بازه اش از تاریخ 1 ژانویه 1753 هست.
      4. به صورت پیش فرض فرمت روبرو رو داره: YYYY-MM-DD HH:MM:SS
   2. ویژگی های DateTime2:
      1. بسته به میزان دقت، 6 تا 8 بایت حافظه اشغال می کنه.
      2. تا 100 نانوثانیه دقت داره.
      3. شروع بازه اش از 1 ژانویه سال 1 میلادی هست
      4. به صورت پیش فرض فرمت روبرو رو داره:   
         YYYY-MM-DD HH:MM:SS[.fractional seconds]
   3. نوع datetime2 جدیدتر و بهینه تر هست. به طور کلی همیشه توصیه میشه ازش استفاده کنیم مگه برای شرایط خاص.
5. تفاوت نوع داده های char, varchar, nchar, nvarchar در چیست؟
   1. Char: طول این نوع داده ثابته؛ براش مهم نیس که به همون مقدار داریم داخلش کاراکتر میذاریم یا کمتر، به هر حال به همون مقداری که برای طولش تعریف شده، حافظه میگیره. شامل unicode ها هم نمی شه پس نمیتونیم بهش مثلا کاراکتر فارسی بدیم.
   2. Nchar: عین char هست اما unicode ها رو هم شامل میشه به همین خاطر به ازای هر کاراکتر، 2 بایت در واقعیت میگیره.
   3. Varchar: این حداکثر طول دیتا رو تعیین می کنه و بر خلاف char، طول دیتا هر مقداری که باشه ( چه برابر با طول تعیین شده چه کمتر از اون) به همون مقداری که کاراکتر وارد شده حافظه میگیره.
   4. Nvarchar: عین varchar اما unicode ها رو هم شامل میشه پس به ازای هر کاراکتر، در واقعیت 2 بایت میگیره.
   5. برتری char و nchar توی سریعتر بودنشونه (چون fixed-size ان و در نتیجه انجین دیتابیس محل دقیق دیتا و طول اون رو میدونه پس سریعتر پیداش میکنه) و برتری varchar و nvarchar توی انعطاف پذیریشون برای تخصیص حافظه اس.
6. Index ها چیستند و بر چند نوع اند؟
   1. یه index یه schema object هست؛ هدفش اینه که سرعت واکشی داده رو بالاتر ببره (گرچه باعث میشه عملیات های insert, update کندتر انجام بشن و مقداری حافظه هم به خاطرش میگیره).
7. تفاوت clustered index و non-clustered index؟
8. یه جدول داریم با 1000 رکورد، اینو index روش میزنیم یا نه و چرا؟
   1. قبلا چون ظرفیت حافظه ها پایین بود و سرعت نداشتن (RAM های ضعیف) اما الان چون هم ظرفیت هم سرعت بالا رفته، دیگه نیاز نیست روی هر جدول index بزنیم و جداول با رکورد های بالا معمولا روشون index میزنن.
9. Stored procedure ها چیستند؟
   1. مجموعه ای از یک یا چند کوئری در SQL هست که قابلیت ذخیره سازی و استفاده مجدد رو داره در نتیجه نیاز به هر بار نوشتن این کوئری ها موقع کال کردن شون نیست. (به اختصار sp هم بهشون میگن)
   2. به Sp ها میشه پارامتر های ورودی داد و همچنین میشه ازش خروجی هم گرفت.
   3. هر sp 2 قسمت داره:
      1. بدنه (definition or body) که شامل قسمت های زیر هست:
         1. نام sp: با دستور CREATE PROCEDURE “procedure-name”  
            اسم sp رو مشخص می کنیم.
         2. پارامتر های ورودی: اگر sp ما پارامتر ورودی داشته باشه، بعد از تعریف اسمش، باید پارامتر های ورودی رو تعریف کنیم که شامل اسم هر پارامتر و datatype اون میشه.  
            نکته: برای پارامتر های خروجی، باید بعد از datatype اش کلمه OUTPUT رو بزاریم.
         3. دستورات مورد نظر: مثل Scope در زبان های برنامه نویسی، با دو keyword به نام های BEGIN و END، جایی که بدنه و دستورات اون sp میخواد نوشته بشه رو مشخص می کنیم و کوئری هامون رو داخل اون مینویسیم.
      2. استفاده (Call): زمانی که میخوایم از اون sp استفاده کنیم، باید از EXEC استفاده کرده و جلوش با ذکر اسم sp مورد نظر و مشخص کردن مقدار هر پارامتر ورودی اون، دستورمون رو تکمیل کنیم.  
         نکته: اگه پارامتر خروجی داشته باشیم، باید حتما با همون کلمه OUTPUT این پارامتر رو هم در زمان call بنویسیم.
   4. بعد از ساخت sp، اگه از داخل خود دیتابیس بازش کنیم، احتمالا می بینیم که یه دستور اضافه ای داره به اسم SET NOCOUNT. این دستور معنیش اینه که موقع اجرا شدن sp، تعداد سطر هایی که تحت تاثیر این sp قرار میگیرن رو بشمره یا نه. اگه دقت کنیم، بعد از اجرای sp، توی قسمت Messages یه سری پیام میاد که بعضی هاشون به این صورته: 10 rows affected. این پیام ها از همون SET NOCOUNT میان که اگه sp این رو ON کنه، دیگه این پیام ها نشون داده نمیشه. به صورت پیش فرض این دستور روی ON قرار داده شده.
   5. یه دستور دیگه ای هم هست به اسم SET ANSI\_NULLS. فرض کنیم که توی دستورات و کوئری هامون داریم از WHERE استفاده می کنیم و یه پارامتر ورودی ای داره داخل یکی از شروط WHERE مون استفاده میشه. اگه این پارامتر ورودی null باشه و  
      ANSI\_NULLS رو ON گذاشته باشیم، اونوقت نتیجه اون شرط، UNKNOWN (NULL)میشه و دیگه داخل نتیجه کوئری نمیاد. جدول زیر حالات مختلف ANSI\_NULLS رو نشون میده:  
      
   6. برای هر sp، میشه مشخص کرد که چه کاربری اجازه دسترسی بهش رو داشته باشه در نتیجه امنیت رو داخل دیتابیس بالا می بره.
   7. زمانی که یه sp رو call می کنیم، صرفا همون اسمش و پارامتر هاش داره به سرور فرستاده میشه در نتیجه ترافیک سرور پایین میاد.
10. تفاوت stored procedure با function؟
    1. Sp میاد یه بار ساخته و ذخیره میشه و همچنین execution plan اش رو همون موقع تعیین می کنه اما function در هر بار استفاده داره execution plan رو می سازه و در نتیجه یه سربار به این خاطر داره.
    2. داخل function فقط میشه DQL نوشت (SELECT) اما sp میتونه DML (INSERT, DELETE, UPDATE) رو هم داشته باشه)
    3. Sp میتونه پارامتر خروجی داشته یا نداشته باشه اما function باید خروجی داشته باشه.
    4. داخل یه sp میشه یه function رو Call کرد اما داخل یه function نمیشه یه sp رو کال کرد.
    5. داخل SELECT, WHERE, HAVING, میشه از function استفاده کرد اما از sp نه.
    6. داخل sp میشه از try/catch استفاده کرد اما داخل sp نه.
    7. داخل sp میشه از transaction استفاده کرد اما داخل function نه.
    8. از temp table میشه داخل sp استفاده کرد اما داخل function نه.
11. Trigger ها چیستند؟
    1. یه SP هایی هستن که وقتی اتفاق خاصی بیفته، به صورت خودکار اجرا میشن.
    2. از اونجایی که قراره وقتی اتفاقی می افته اجرا بشن، در اصل بعد یا قبل از دستورات DML (INSERT, UPDATE, DELETE) اجرا میشن.
    3. برای هر جدول، 6 نوع trigger میشه تعریف کرد: AFTER INSERT, AFTER UPDATE, AFTER DELETE, BEFORE INSERT, BEFORE UPDATE, BEFORE DELETE
    4. میتونه روی جدول، رکورد یا فیلد باشه.
    5. برای جدول تعریف میشه نه دیتابیس.
12. Cursor ها توی stored procedure چیستند؟
13. انواع JOIN ها؟
    1. INNER JOIN:
    2. LEFT JOIN:
    3. RIGHT JOIN:
    4. FULL OUTER JOIN:
    5. CROSS JOIN:
14. تفاوت Hash, merge, loop در join ها چیست؟
    1. اینا نحوه انجام join هستن؛ اگه نذاریمشون، خود انجین sqlserver یکی از اینا رو انتخاب میکنه.
15. دستور COMMIT؟
    1. وقتی یه select انجام میشه، همون موقع نتیجه اش میاد و میره روی مموری اما هنوز روی دیسک ذخیره نشده؛ با COMMIT میاد اون نتیجه رو (از آخرین COMMIT تا این COMMIT) از مموری میبره روی دیسک ذخیره میکنه.
16. تفاوت where و having؟
    1. Where میاد قبل از اینکه group by انجام بشه شرط رو اعمال میکنه اما having بعد از group by اعمال میشه.
    2. Where بدون group by هم کار میکنه اما having باید حتما با group by بیاد.
    3. توی where نمیشه از aggregagte function ها استفاده کرد اما having میتونه.
17. دستور UNION؟
    1. میاد دوتا select که تعداد ستون های برابر و نوع های ستون ها هم برابر هستن رو با هم ترکیب میکنه (مثل اجتماع دو مجموعه)
18. تفاوت UNION و UNION ALL؟
    1. توی UNION اگه دیتایی duplicate شده باشه، توی خروجی نهایی حذفش میکنه اما UNION ALL میگه هیچ اهمیتی نداره که duplicate وجود داشته باشه و هرکوئری رو به صورت مستقل اجرا میکنه و توی خروجی اضافه میکنه.
19. دستور SUBTRACT؟
    1. مثل union اما برعکس. میاد حاصل select دوم رو از اولی کم میکنه (تفریق مجموعه ها).
20. دستور DISTINCT؟
    1. رکورد های duplicate رو حذف میکنه.
21. تفاوت SQL و T-SQL؟
    1. T-SQL در واقع یه extension از SQL به حساب میاد که بیشتر برای MSSQLServer به کار میره؛ در واقع همون SQL هست اما با یه سری چیزای اضافه تر.
    2. SQL به صورت open source هست اما T-SQL توسط خود مایکروسافت توسعه پیدا میکنه.
    3. دستورات اضافه تر یا به صورت متفاوت با SQL در T-SQL پیاده سازی شدن:
       1. دستور LIMIT در SQL با TOP در T-SQL برابره
       2. دستور ISNULL در T-SQL که اگر هنگام کوئری گرفتن، با NULL در دیتا مواجه بشه، به جاش یه مقدار مد نظر ما رو قرار میده.
22. کامند EXPLAIN ANALYZE؟
23. CAP theorem؟
    1. یه قاعده ای هست که میگه هیچ دیتابیسی نمیتونه همزمان سه ویژگی زیر رو داشته باشه (در بهترین حالت 2 ویژگی رو همزمان داره):
       1. Consistency: این ویژگی یعنی تمام node های دیتابیس توی شبکه باید در آن واحد یه مقدار برابر رو برگردونن. این مقدار، آخرین مقدار درستِ آپدیت شده اس. اگه مقدار های توی یه node با یک یا چندتا node دیگه یکی نباشه، اون دیتابیس inconsistent هست.
       2. Availability: هر درخواستی که به دیتابیس میاد، باید یا کامل انجام بشه (در یه مدت زمان مشخص) یا اصلا انجام نشه (پیامی مبنی بر انجام نشدن بده). این قضیه برای تمام node های دیتابیس صدق میکنه؛ یعنی باید همه شون درخواست رو کامل انجام بدن تا عملیات موفق تلقی بشه و بتونیم بگیم که دیتابیس ما این ویژگی رو داره. البته که ممکنه این ویژگی همیشه برقرار نباشه.
       3. Partition tolerance: این ویژگی میگه که اگه به هر دلیلی یه node دیتابیس به مشکل بخوره و نتونه کار بکنه، باقی node ها بتونن جور اش رو بکشن؛ یعنی از دست رفتن یه node مشکلی در کار ایجاد نکنه.
    2. حالا حالت های مختلف دوتایی از این ویژگی ها رو بررسی میکنیم:
       1. CA (consistency, availability):
          1. نمونه: mySQL, postgreSQL
       2. AP (availablity, partition tolerance):
          1. نمونه: dynamo DB, google cloud spanner, cassandra db
       3. CP (consistency, partition tolerance):
          1. نمونه: Hbase, mongoDB, redis
       4. 
24. Normalization و سطوح مختلف اون؟
25. Rank و row number؟
26. Function های sql؟
    1. AVG, MAX, MIN, SUM, COUNT, FIRST, LAST
    2. UCASE, LCASE, MID, LEN, ROUND
    3. MID, LEN, NOW, FORMAT
27. Function vs stored procedure؟
28. منظور از sargable query چیه؟
    1. از index ها استفاده میکنن
    2. فانکشن های خود SQL مستقیم روی ستون ها اعمال میشن در نتیجه از index ها استفاده نمیکنن اما sargable query اگه باشه، روی index اعمال میکنه.
    3. توی قسمت WHERE، از function ها یا محاسبات روی ستون های index شده باید دوری کرد.
    4. تا جایی که میشه مقایسه ها رو مستقیم انجام بدیم، به جای گذاشتن ستون توی function ها.
29. Filepage چیست؟
30. Execution plan چیست؟
31. Temp table ها چیستند؟
32. آیا می توان در runtime دیتابیس را عوض کرد؟
33. یه سری از خونده های قبلیت توی Onenote هست