31일차

DP 900

정규화는 데이터베이스 테이블들 간의 관계를 제거한다? -> X

정규화는 데이터의 중복이 줄어든다? -> O

정규화는 데이터의 무결성이 향상된다? -> O

추출, 변환, 로드(ETL) 과정은 OO가 필요하다?

* 데이터를 전환하기에 충분한 파워가 있는 데이터 스토어(저장소)가 필요하다

일괄 처리(batch process)에서 -> 지연시간(latency) 발생이 예상된다

오디오 파일 전사(transcribing – 받아쓰기)는 -> cognitive analytics의 예이다

**이건 조금 외워야한다…**

지난 달의 판매가 왜 증가했나? -> Diagnostic

다른 상품을 구입할 때 내 예산을 어떻게 할당해야 하나? -> Prescriptive

어떤 사람들이 회사 문서에 언급되었는가? ->Descriptive

텍스트이(가) 표시된 사진

자동 생성된 설명

Customer는 a root object

Address는 a nested object

Social media는 a nested array -> []로 묶여 있으니까 배열

텍스트이(가) 표시된 사진

자동 생성된 설명

데이터 모델은 -> star schema -> 별 모양처럼 사방으로 붙어있는 형태

Customer는 -> dimension

애저 시냅스 애널리틱스의 Massively parallel processing(MPP) 엔진은

cf) parralle: 병렬 <-> serial: 직렬,

여기서 말하는 것은 여러 컴퓨터들을 병렬 연결하여 하나로 집결하여 처리하는 프로세스를 말함

* 컴퓨팅 노드들에게 프로세스를 배포하는 기능을 말한다.

OO은 테이블의 키값을 기준으로 그것들을 축소하여 정렬해두는 object를 말한다.

* a clustered index

관계형 데이터베이스는 방대한 양의 OO 시나리오에 적합하다

* transactional writes(쓰기 지연?)

일괄 처리는 file store에 데이터를 출력할 수 있다 -> O

관계형 데이터베이스에 출력할 수 있다 -> X

NoSQL 데이터베이스에 출력할 수 있다. -> X

ELT process 중 Extract -> CRM system(고객 데이터)에서 추출

Load: -> data warehouse 에 로드

Transform i-> data integration tool로 변환

사각형들이 각각의 크기와 색상들로 개별 데이터들의 상대적 지표를 나타내는 차트

* 트리맵

선택한 결과에 따라 주요 기여도를 표시하는 차트

* 키 인플루언서

두 숫자 지표들 간의 관계를 보여주는 차트

* 스캐터(Scatter)

애저 지역 외부로 자동으로 복제되는 스토리지를 원한다. 어떤 스토리지 계정을 만들어야 하는가?

* region 바깥으로 나가려면, ‘geo’가 포함되어야 한다!! -> RA-GRS, GRS

PaaS데이터베이스는 세팅과 설정에 IaaS보다 덜 수고롭다 -> O

PaaS는 OS버전을 컨트롤하고 업데이트 할 수 있다. -> X – PaaS는 그냥 제공되는 걸 쓴다

PaaS기반의 관계형, 비관계형 데이터베이스는 비용을 줄이기위해 일시 정지할 수 있다 -> X

* 데이터베이스는 한번 구동하면 멈출 수 없다… ㅠ

DML에 해당하는 쿼리는? -> INSERT

customer data와 order data를 combine했다(JOIN 했다)! 다른 이들도 같은 쿼리를 할 수 있게 하려면?

* a view - view객체를 만들어두면 같은 쿼리를 여러 번 할 때에 편하다

key/value 형태의 데이터는 OO에 적합하다

* simple lookups -> key/value store는 단순한 쿼리를 빠르게 진행할 경우 사용

Azure Blob storage -> image files – blob은 아무거나 다 들어가니까 남는거

Azure Cosmos DB Gremlin API -> Relationships between employees – Gremlin은 관계도!

Azure Table storage -> key/value pairs – 테이블이니까 키/밸류

Azure Resource Group

* Azure Storage account
  + File share
    - Folders
      * Files
* Container가 들어가려면 Blob이 먼저 나와야 한다!

real-time-processing에 대한 두가지 옳은 것은?

* Low latency is expected(짧은 대기시간)
* Data is processed as it is created(데이터 즉시 처리)

데이터 저장공간 안의 데이터 구조를 설명하는 것은? -> Dataset

외부 리소스와 연결하기 위해 사용되는 정보 -> Linked service

특정한 작업 단위를 수행하고 일정을 잡을 수 있는 논리적인 그룹? -> Pipeline – 데이터가 흐르는 라인 그 자체를 말함

12시간마다 물품 카탈로그를 로드 -> batch

수천개의 온라인 구매를 리얼타임으로 처리 -> Streaming

백만번의 상호작용마다 데이터 업데이트 -> batch

오디오 파일의 closed caption text를 만드는 것은 OO analytics의 예이다

* descriptive(설명)

대학의 학과별 학생 현황을 시각화 하는 것은 OO ananlytics의 예이다

* descriptive

Transparent Data Encryption(TDE: 투명한 데이터 암호화)는 OO이다

TDE? – 자료를 넣으면 자동으로 암호화되고 꺼내면 자동으로 복호화 되는 기능

* 난 잘 모르겠는데 지가 알아서 암호화, 복호화 한다!
* 데이터베이스를 암호화해서 미사용 데이터를 보호한다.

다중 인증을 사용하여 Azure SQL database에 로그인하려면 어떤 인증을 사용해야 하는가?

* AAD(Azure Active Directory)를 사용한다

Azure SQL database와 비교해서 Azure SQL managed instance를 사용할 때의 장점이 무엇인가?

* 여러 데이터베이스들에 쿼리와 트랜잭션을 걸 수 있다!
* cross-database queries and transaction

Azure SQL database를 기본적으로 보호하는 장치는?

1. NSG(Network security group -> VM에 달려있는 것
2. **sever-level firewall**
3. Azure Firewall -> 애저 서비스 전체 앞에 위치
4. Azure Front Door -> 상관없음;

GUI가 있는 데이터베이스를 만들려고 한다. 프로젝트 기반의 오프라인 데이터베이스를 만든다면, 어떤 도구가 좋을까?

* SSDT(Microsoft SQL Server Data Tools)

다른 네트워크에서 Azure SQL database에 접근하는 것을 막는다 -> Firewall

Azure SQL database에 로그인할 때 AADfmf 돕는 것 -> Authentication

민감 정보가 일반 텍스트로 보이지 않게 하는 것 -> 암호화

Azure SQL managed instance를 이용해서 상호작용하는 app을 만들고자 한다. Read-only database replica를 만들어야 하는 이유로 적절한 것은?

* transactional workload에 영향을 주지 않으면서 보고서를 만들어야 할 경우
* 보고서를 쓰려면 전체 데이터에 접근해야 하는데, 그러면 다른 사람들의 업무가 불가…
* 이런 일을 막기 위해 읽기 전용으로 복제를 만든다!

관계형 데이터베이스는 OO 할 때 써야한다.

* 강력한 일관성(Consistency)가 보장되어야 할 때에 사용한다

관계형 데이터는 OO을 이용해서 다른 데이터 테이블과 관계성을 강화한다

* 키

OO은 쿼리문을 통해 얻은 내용들을 포함하는 가상의 테이블이다.

* View

Azure SQL database에서 인터넷을 이용하여 Products 테이블에 쿼리를 하려고 한다. 뭐가 필요한가?

* 해당 테이블에 대한 SELECT 권한이 있어야 한다!
* 데이터베이스에 유저가 있어야 한다
* IP 주소가 데이터베이스에 접속할 수 있도록 허용되어 있어야 한다.

관계형 데이터베이스를 사용하려고 하는데, 투입할 노력을 최소화하려면 어떤 데이터베이스를 써야하는가?

* Azure SQL database(PaaS니까 관리비용 가장 적게 든다)

Azure SQL database에서 쿼리문의 결과를 SQL notebook 형태로 쓰려고 한다. 어떤 기능이 필요한가?

* Azure Data studio

데이터베이스의 객체 중 실제로 데이터를 가지고 있는 것? -> 테이블

쿼리문으로 내용이 정해지는 것? -> 뷰

데이터 검색 속도를 빠르게 할 수 있는 것? -> 인덱스

e-commerce app을 만들 것이라면 어떤 어플리케이션을 쓰는게 좋을까?

* Online Transaction Processing(OLTP)
* 어플리케이션 단위에서는 OLTP가 중요하다

Azure Resource Manager template을 언제 쓸까?

* 반복적으로 상호 종속적인 리소스 그룹을 만드는 것을 자동으로 할 때
* 템플릿에 각각이 상호 종속적인 부분을 기록하여 던지면 한 뭉텅이 씩 만들어짐!

인터넷으로 접속하는 Azure SQL database에서 IP주소가 바뀐 다음 데이터베이스에 접속이 안된다면, 어디가 문제인가?

* a database-level firewall

SQL 서버를 관리하거나 SQL database에서 접속,설정 매니지 등을 돕는 그래픽 툴은?

-> SSMS(Microsoft SQL Server Management Studio) -> 이게 가장 기능이 많음

아주 가벼운 소스코드 에디터이며 SQL Server로 연결하거나 T-SQL에 풍부한 에디팅 경험을 제공하는 툴은?

-> Visual Studio Code

아주 가벼운 에디터이며 JSON, Excel파일로 SQL 쿼리 결과를 저장할 수 있는 툴은?

-> Azure Data Studio

관계형 데이터베이스의 테이블에는 columns와 rows를 포함한다 -> O

인덱스는 테이블에서 데이터 타입을 설명한다 -> X

뷰는 쿼리문으로 만들어진 가상 테이블이다 -> O

Azure SQL database의 커맨드-라인 툴로 어떤 것을 쓸 수 있나?

* sqlcmd

Azure SQL Database는/ 관리받는 백업 서비스가 있다 -> O

고가용성이 포함되어 있다 -> O

Azure Defender를 사용할 수 있다 -> O, 그러나 유료

MySQL과 MariaDB는 내부적으로 동일하므로, MySQL 워크벤치로 MariaDB 데이터베이스에 쿼리할 수 있다!

PaaS 데이터베이스는/ 고가용성이 포함되어 있다? -> O

확장 옵션이 포함되어 있다? -> O

하드웨어 관리하는 오버헤드를 줄여준다? -> O, PaaS니까 관리비용 줄여준다

Azure SQL database를 사용하면/ OS 업데이트를 주기적으로 해야 한다? -> X

Microsoft 365 구독이 필요하다? -> X

비용을 줄이기 위해 기존에 가지고 있는 라이선스를 사용할 수 있다? -> O

Azure Data Studio는 MacOS에서 실행이 가능한가? -> O

Microsoft SQL Server Management Studio(SSMS)는 SQL notebooks 사용이 가능한가?

-> X, Data Studio만 가능!

Azure Data Studio는 데이터베이스 복원이 가능한가? -> O

PaaS 관계형 데이터베이스를 이용할 때의 이점이 무엇인가?

* 최신 버전에 엑세스
* 인프라 관리비용 감소

애저에서 PaaS 데이터베이스를 사용한다면/ OS 업데이트를 직접 해야한다? -> X

자동으로 백업 된다> -> O

DB 엔진을 직접 업데이트 해야한다? -> X

Azure SQL database를 만들었을 때, 데이터베이스에 항상 연결할 수 있는 계정은?

* 논리 서버의 서버 어드민 계정

개발 중에 Win10과 Ubuntu 두가지 컴퓨터가 있다. 인텔리센스가 다 지원되고 Azure SQL database에서 작업하려면 어떤 도구를 써야하나?

* Azure Data Studio

윈도우 서버에 공유 폴더를 만들고 Azure Storage에 옮기려고 한다. 어떤 스토리지를 쓸까?

* files

쇼핑카트를 데이터베이스로 만들 때, 검색할 때에 가장 작은 대기시간을 가지는 데이터 타입이 무엇이겠냐?

* columnar: 테이블 형태의 데이터(RDB)가 속도는 제일 빠르다!

윈도우에서 구동할 앱을 개발 중, 맵핑 되어있는 드라이브에 접근할 수 있어야 한다. 어떤 기능을 쓸까?

* Azure Files

애저 Cosmos DB는/ API별로 따로 설정해줘야 한다?

-> X, 분리하는게 좋긴 한데… 합쳐서 쓸 수 있다

최적화 할 때에 파티션 키를 쓴다? -> O

같은 Cosmos DB의 논리적 파티션은 다른 파티션 키를 가질 수 있다? -> X, 그럼 안되지;

큰 용량의 JSON 데이터를 다루려고 하고, app이 구조를 정의할 수 있다. 어떤 데이터 스토어를 사용할까?

* key/value

비관계형 데이터를 저장하고, 텍스트, 비디오, 오디오 등의 파일을 저장하려고 한다. 어떤 데이터 스토어가 좋겠는가?

* object, 파일 자체를 다루는건 object -> 그냥 저장소(storage)에 가까움

인터넷에 연결된 온도 센서를 저장하려고 한다. 데이터는 감지된 온도의 trends가 될 것이다. 어떤 데이터 스토어를 사용할 것인가?

* time series

Azure Data Lake Storage로 Azure region을 넘어 데이터를 수집하면 대역폭 비용이 발생한다?

-> O, 데이터가 넘어와야 해서 비용 발생

동일한 계정 하나 밑에 blob, table, file storage를 다 만들 수 있나? -> O

Azure Data Lake Storage를 구축하는데 Azure Storage 계정 밑에 만들 수 있나? -> O