

Assignment 3st June 2022

NAME: **ASHOK KUMAR**

ROLL NUMBER: DXC-262-AB-1233

BATCH: DXC-262-ANALYTICS-B12-AZURE

COMPANY: DXC TECHNOLOGY

EMPLOYEE DOMAIN: AZURE ANALYTICS

TRAINING UNDER: MANIPAL PRO LEARN

TRAINER NAME: MR. AJAY KUMAR

DATE OF SUBMISSION: 3 JUNE 2022

NO. OF QUES: 12

QUESTION 1. Explain various Difference between SQL & NoSQL DBs?

ANSWER:

1. SQL databases are relational, NoSQL databases are non-relational.
2. SQL databases use structured query language and have a predefined schema. NoSQL databases have dynamic schema for unstructured data.
3. SQL databases are vertically scalable and NoSQL databases are horizontally scalable.
4. SQL databases are table-based and NoSQL databases are document, key-value, graph and wide-column stores.
5. SQL databases are better for multi-row transactions and NoSQL is better for unstructured data like documents or JSON.
6. SQL databases are best suited for complex queries. NoSQL databases are not so good for complex queries.
7. SQL databases are not best suited for hierarchical data storage but NoSQL databases are best suited for hierarchical data storage.
8. MySQL, Oracle, Sqlite, PostgreSQL and MS-SQL etc are the example of SQL database.
MongoDB, BigTable, Redis, RavenDB etc are the example of nosql database.

QUESTION 2. Explain advantages of NoSQL DBs? Explain how MongoDB data will be inserted?

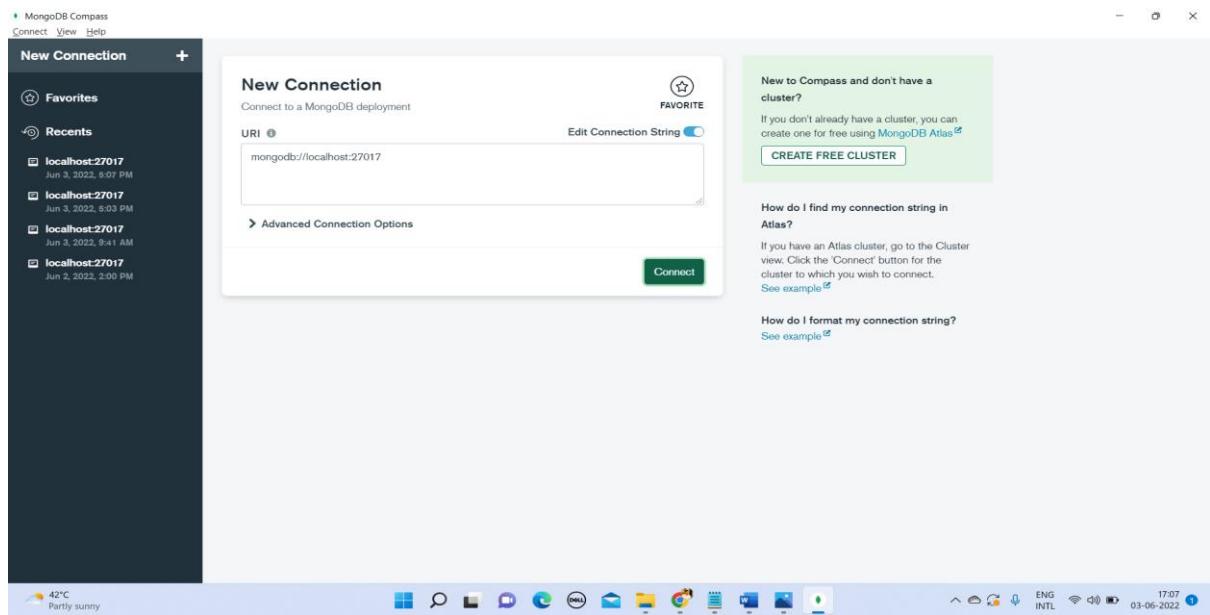
ANSWER:

Advantages of NoSQL database:

- **Scalable and highly available:** Many NoSQL databases are designed to support seamless, online horizontal scalability without significant single points of failure.
- **Flexible data models:** Most non-relational systems do not require developers to make up-front commitments to data models. Existing schemas are dynamic, so they can often be changed "on the fly."
- **Dynamic schema for unstructured data:** Documents can be created without a defined structure first, which enables each to have its own unique structure. Syntax varies per database and fields can be added as you build the document.
- **High performance:** A limited database functionality range (e.g., by relaxing durability guarantees) enables high performance amongst many NoSQL databases.
- **High-level data abstractions:** Beyond the "value in a cell" data model, NoSQL systems provide high-level APIs for powerful data structures. For example, Redis includes a native-sorted set abstraction.

Insert data into MongoDB:

Connect to a MongoDB deployment



MongoDB Compass - localhost:27017

Connect View Help

localhost:27017

5 DBS 7 COLLECTIONS C

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

DZXDemoDB1

admin

config

local

test

> MONGOSH

42°C Partly sunny

My Queries Databases Performance

Create database View

Sort by Database Name

admin

Storage size: 20.48 kB Collections: 1 Indexes: 1

config

Storage size: 24.58 kB Collections: 1 Indexes: 2

DZXDemoDB1

Storage size: 11.45 MB Collections: 1 Indexes: 1

local

Storage size: 45.06 kB Collections: 1 Indexes: 1

test

Storage size: 81.92 kB Collections: 4 Indexes: 5

This screenshot shows the MongoDB Compass interface on a Windows desktop. The left sidebar lists databases: admin, config, DZXDemoDB1, local, and test. The main area displays the 'Databases' tab with a table for each database showing storage size, collection count, and index count. A 'Create database' button is at the top. The taskbar at the bottom shows various application icons.

Click on create database and type database name and collection name.

MongoDB Compass - localhost:27017

Connect View Help

localhost:27017

5 DBS 7 COLLECTIONS C

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.9 Community

My Queries

Databases

Filter your data

DZXDemoDB1

admin

config

local

test

> MONGOSH

42°C Partly sunny

My Queries Databases Performance

Create database View

Sort by Database Name

admin

Storage size: 20.48 kB Collections: 1 Indexes: 1

config

Storage size: 24.58 kB Collections: 1 Indexes: 2

DZXDemoDB1

Storage size: 11.45 MB Collections: 1 Indexes: 1

local

Storage size: 45.06 kB Collections: 1 Indexes: 1

test

Storage size: 81.92 kB Collections: 4 Indexes: 5

Create Database

Database Name: College

Collection Name: Students

Advanced Collection Options (e.g. Time-Series, Capped, Clustered collections)

Cancel Create Database

This screenshot shows the 'Create Database' dialog box in the foreground. It has fields for 'Database Name' (College) and 'Collection Name' (Students). The background shows the same MongoDB Compass interface as the first screenshot, with the 'Databases' tab selected. The taskbar at the bottom is visible.

The screenshot shows the MongoDB Compass interface. On the left, the sidebar lists databases (localhost:27017, DZXDemoDB1) and collections (admin, config, local, test). The main area is titled 'College.Students' and shows a single document entry. The status bar at the bottom indicates 'MONGOSH' and the system tray shows the date and time as 03-06-2022.

We can import data from a JSON or CSV file.

The screenshot shows the MongoDB Compass interface with the 'College.Students' collection now containing four documents. Each document has fields: '_id', 'id', 'name', 'occupation', and 'emp_age'. The names listed are Raneesh, Sureesh, Vijay, and Akshay. The status bar at the bottom indicates 'MONGOSH' and the system tray shows the date and time as 03-06-2022.

Also we can use Command Prompt to insert data.

Go to **C:\Program Files\MongoDB\Server\5.0\bin** and open cmd.

Run these-

C:\Program Files\MongoDB\Server\5.0\bin>mongod

C:\Program Files\MongoDB\Server\5.0\bin>mongo

```

C:\Windows\System32\cmd.exe - mongo
[{"date": "2022-06-03T17:17:48.088+05:30"}, {"s": "i", "c": "-", "id": 20520, "ctx": "initandlisten", "msg": "Stopping further Flow Control ticket acquisitions."}, {"date": "2022-06-03T17:17:48.089+05:30"}, {"s": "i", "c": "NETWORK", "id": 4784910, "ctx": "initandlisten", "msg": "Shutting down the ReplicaSetMonitor"}, {"date": "2022-06-03T17:17:48.090+05:30"}, {"s": "i", "c": "JOURNALING", "id": 4784921, "ctx": "initandlisten", "msg": "Shutting down the MigrationUtilExecutor"}, {"date": "2022-06-03T17:17:48.091+05:30"}, {"s": "i", "c": "ASYNCREPLICATOR", "id": 4784922, "ctx": "initandlisten", "msg": "Killing any outstanding egress activity."}, {"date": "2022-06-03T17:17:48.092+05:30"}, {"s": "i", "c": "COMMAND", "id": 4784923, "ctx": "initandlisten", "msg": "Terminating the serviceEntryPoint"}, {"date": "2022-06-03T17:17:48.093+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784925, "ctx": "initandlisten", "msg": "Shutting down free monitoring"}, {"date": "2022-06-03T17:17:48.094+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784926, "ctx": "initandlisten", "msg": "Shutting down the HealthLog"}, {"date": "2022-06-03T17:17:48.095+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784928, "ctx": "initandlisten", "msg": "Shutting down the TTL monitor"}, {"date": "2022-06-03T17:17:48.096+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784929, "ctx": "initandlisten", "msg": "Acquiring the global lock for shutdown"}, {"date": "2022-06-03T17:17:48.097+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784931, "ctx": "initandlisten", "msg": "Dropping the scope cache for shutdown"}, {"date": "2022-06-03T17:17:48.098+05:30"}, {"s": "i", "c": "CONTROL", "id": 4784926, "ctx": "initandlisten", "msg": "Shutting down full-time data capture"}, {"date": "2022-06-03T17:17:48.099+05:30"}, {"s": "i", "c": "CONTROL", "id": 20565, "ctx": "initandlisten", "msg": "Now exiting"}, {"date": "2022-06-03T17:17:48.099+05:30"}, {"s": "i", "c": "CONTROL", "id": 23138, "ctx": "initandlisten", "msg": "Shutting down", "attr": {"exitCode": 100}}
C:\Program Files\MongoDB\Server\5.0\bin\mongo
MongoDB shell version v5.0.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongod
Implicit session: session { _id : UUID('93be166c-e1fa-4334-bc08-ac27594df099') }
MongoDB server version: 5.0.9
=====
Warning: the "mongo" shell has been superseded by "mongosh", which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in an upcoming release.
For installation instructions, see https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
2022-06-03T12:09:25.618+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
=====
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
=====
42°C Party sunny ENG INTL 17:38 03-06-2022

```

Show dbs will show all databases.

Show collections will show all collections.

Use College will take you to college database.

db.teachers.insert({id:"123",name:"akash",subject:"computer science"}); will insert data into teachers collections of College database.

db.teachers.find(); will show data of teachers collection.

```

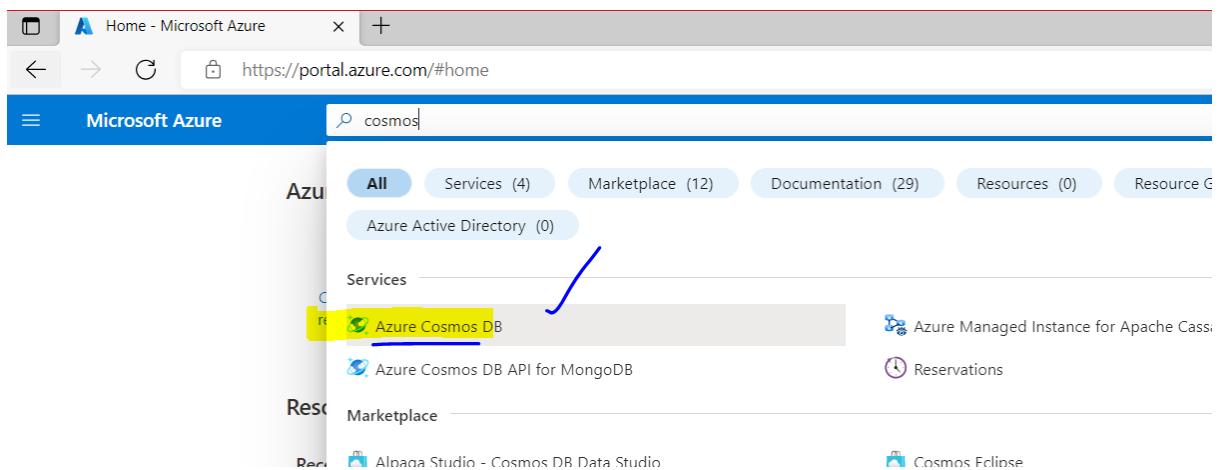
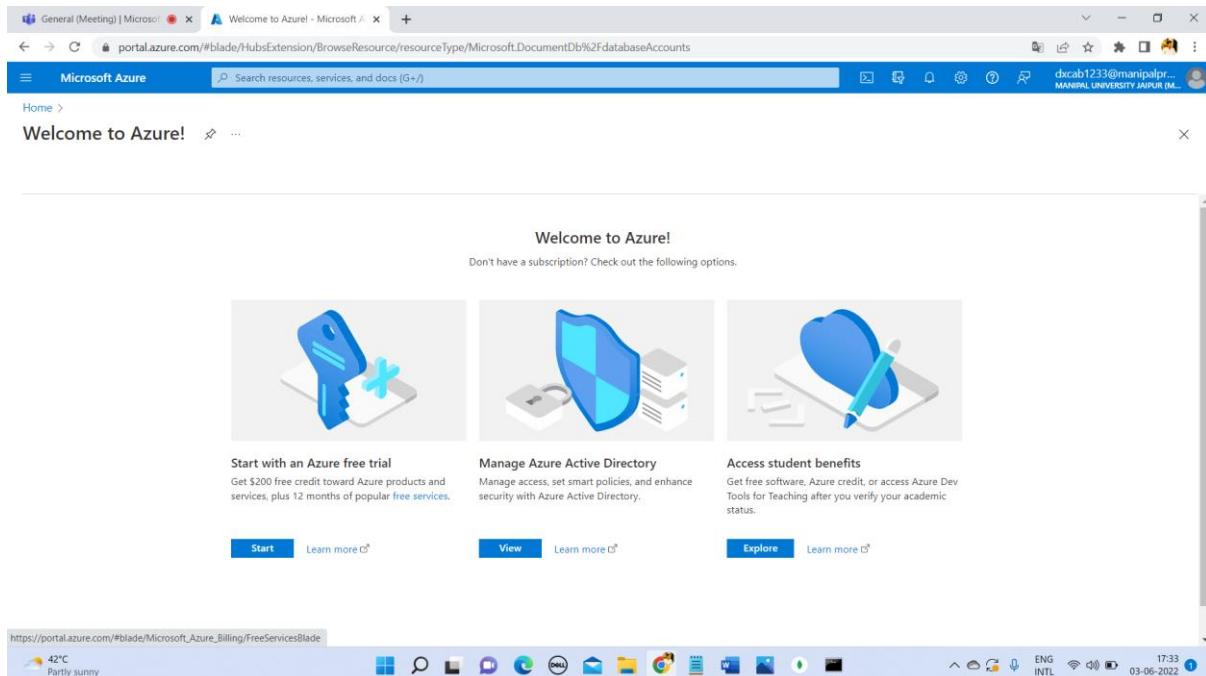
MongoDB Compass - localhost:27017/College.teachers
Connect View Collection Help
localhost:27017
DBS COLLECTIONS FAVORITE
HOST localhost:27017
CLUSTER Standalone
EDITION MongoDB 5.0.9 Community
My Queries
Databases
Filter your data
College
Students
teachers
DZXDemo01
admin
config
local
test
COLLEGE.teachers
Documents Aggregations Schema Explain Plan Indexes Validation
FILTER { field: 'value' }
ADD DATA VIEW
_id: ObjectId('6299f652e3a904dd72290bf9')
id: "123"
name: "akash"
subject: "computer science"
_id: ObjectId('6299f709f3a904dd72290bf9')
id: "123"
name: "akash"
subject: "computer science"
_id: ObjectId('6299f709f3a904dd72290bf9')
id: "124"
name: "rakesh"
subject: "computer science"
C:\Windows\System32\cmd.exe - mongo
> show dbs
College 0.020GB
DZXDemo01 0.021GB
admin 0.000GB
config 0.000GB
local 0.000GB
test 0.000GB
> show collections
Students
teachers
> use College
switched to db College
> db.teachers.insert({id:"123",name:"akash",subject:"computer science"});
WriteResult({ nInserted: 1 })
> db.teachers.find();
( { _id: ObjectId('6299f652e3a904dd72290bf8'), "id": "123", "name": "akash", "subject": "computer science" },
{ _id: ObjectId('6299f6af3a904dd72290bf9'), "id": "123", "name": "akash", "subject": "computer science" } )
> db.teachers.insert({id:"124",name:"rakesh",subject:"computer science"});
WriteResult({ nInserted: 1 })
> db.teachers.find();
( { _id: ObjectId('6299f652e3a904dd72290bf8'), "id": "123", "name": "akash", "subject": "computer science" },
{ _id: ObjectId('6299f6af3a904dd72290bf9'), "id": "123", "name": "akash", "subject": "computer science" },
{ _id: ObjectId('6299f709f3a904dd72290bf9'), "id": "124", "name": "rakesh", "subject": "computer science" } )
>
42°C Party sunny ENG INTL 17:28 03-06-2022

```

Data is inserted into college.teachers

QUESTION 3: Explain the steps - how COSMOS DB can be created with screens?

ANSWER: Go to <https://portal.azure.com/#home>



Search Azure Cosmos DB and click **Add**. This will bring you to a new Create Azure Cosmos DB Account window. Provide in detail like the Resource Group, Account Name, API – Core (SQL), Location, etc. and click on **Review + create**.

Azure Cosmos DB - Microsoft Azure

Microsoft Azure

Home > Azure Cosmos DB

Default Directory

+ Create | Restore | Manage view | Refresh | Export to CSV | Open query | Assign tags

Filter for any field... Subscription == all Resource group == all Location == all Add filter

Name ↑↓ Status ↑↓ Subscription ↑↓



Select API option - Microsoft Azure

Microsoft Azure

Home > Azure Cosmos DB > Select API option

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Core (SQL) - Recommended (highlighted)

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

[Create](#) [Learn more](#)

Azure Cosmos DB API for MongoDB

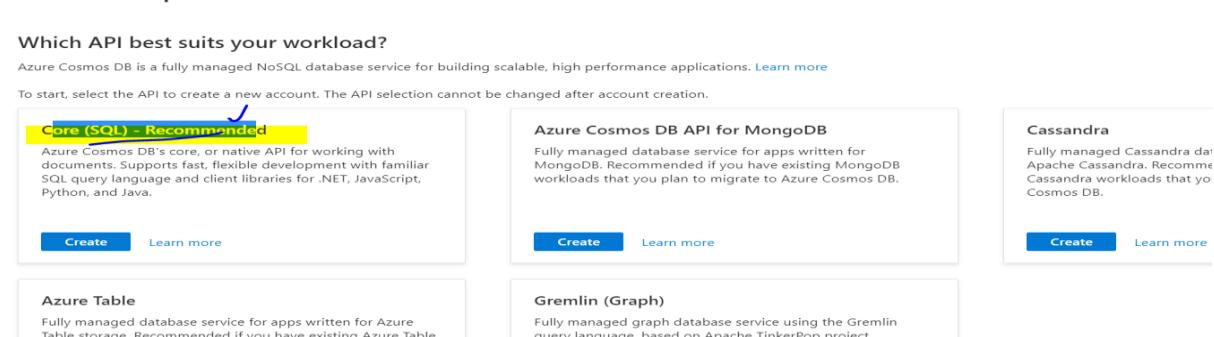
Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Cassandra

Fully managed Cassandra database. Apache Cassandra. Recommended for Cassandra workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)



Create Azure Cosmos DB Account - Microsoft Azure

Home > Azure Cosmos DB > Select API option > Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL database service for building scalable, high performance applications. [Try it for free](#), for 30 days with unlimited resources.

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * [Create new](#)

Resource Group * [Create new](#)

Instance Details

Account Name * [Create new](#)

Location * [Create new](#)

Capacity mode Provisioned throughput Serverless [Learn more about capacity mode](#)

[Review + create](#) [Previous](#) [Next: Global Distribution](#)



Create Azure Cosmos DB Account

https://portal.azure.com/#create/Microsoft.DocumentDB

Microsoft Azure

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Global Distribution

Configure global distribution and regional settings for your account. You can also change these settings after the account is created.

Availability Zones Enable Disable

Review + create Previous Next: Networking

Create Azure Cosmos DB Account

https://portal.azure.com/#create/Microsoft.DocumentDB

Microsoft Azure

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Network connectivity

You can connect to your Cosmos DB account either publically, via public IP addresses or service endpoints, or privately, using a private endpoint.

Connectivity method All networks Public endpoint (selected networks) Private endpoint

All networks will be able to access this CosmosDB account. [Learn More](#)

Review + create Previous Next: Backup Policy

Create Azure Cosmos DB Account

https://portal.azure.com/#create/Microsoft.DocumentDB

Microsoft Azure

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB provides two different backup policies. You will not be able to switch between backup policies after the account has been created. Learn more about the [Backup policies](#).

Backup policy Periodic Continuous

Backup interval 240 60-1440 Minute(s)

Backup retention 8 8-720 Hours(s)

Copies of data retained 2

Backup storage redundancy * Geo-redundant backup storage Zone-redundant backup storage Locally-redundant backup storage

Review + create Previous Next: Encryption

Create Azure Cosmos DB Account X +

Microsoft Azure https://portal.azure.com/#create/Microsoft.DocumentDB

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL) ...

Basics Global Distribution Networking Backup Policy **Encryption** Tags Review + create

Data Encryption

Azure Cosmos DB encryption protects your data at rest by seamlessly encrypting your data as it's written in our datacenters, and automatically decrypting it for you as you read it.

By default your Azure Cosmos DB account is encrypted at rest using service-managed keys. At the moment, you will not be able to switch back to service-managed key if you choose customer-managed keys.

Data Encryption

Service-managed key
 Customer-managed key (Enter key URI)

Review + create Previous **Next: Tags**

Create Azure Cosmos DB Account X +

Microsoft Azure https://portal.azure.com/#create/Microsoft.DocumentDB

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL) ...

Basics Global Distribution Networking Backup Policy **Encryption** **Tags** Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources.

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Key	Value
<input type="text"/>	<input type="text"/>

Review + create Previous **Next: Review + create**

Create Azure Cosmos DB Account

https://portal.azure.com/#create/Microsoft.DocumentDB

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Azure Cosmos DB > Select API option >

Create Azure Cosmos DB Account - Core (SQL)

Validation Success

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Creation Time

Estimated Account Creation Time (in minutes) 2

The estimated creation time is calculated based on the location you have selected

Basics

Subscription	Azure Pass - Sponsorship
Resource Group	dxcrg231
Location	East US
Account Name	(new) dxcosmosdb1
API	Core (SQL)
Capacity mode	Serverless
Availability Zones	Disable

Backup Policy

Backup policy	Periodic
Backup storage redundancy	Geo-redundant backup storage

Networking

Connectivity method	All networks
---------------------	--------------

Create Previous Next Download a template for automation

Microsoft.Azure.CosmosDB-20220602095741 | Overview

Deployment

Search (Ctrl+ /) Delete Cancel Redeploy Refresh

We'd love your feedback! →

Your deployment is complete

Deployment name: Microsoft.Azure.CosmosDB-20220602095741
Subscription: Azure Pass - Sponsorship
Resource group: dxcrg231

Start time: 6/2/2022, 9:57:51 AM
Correlation ID: 79addb37-73dd-402a-8905-41a44a450515

Deployment details (Download)
Next steps

Go to resource

QUESTION 4: Explain how to write JSON query in COSMOS DB?

ANSWER:

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'dxccosmosdb1 - Microsoft Azure' and contains the following navigation items:

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Quick start** (highlighted with a yellow box)
- Notifications
- Data Explorer

The main content area is titled 'dxccosmosdb1 | Quick start' and displays the following steps:

- Step 1: Add a container**

In Azure Cosmos DB, data is stored in containers.

Create 'Items' container

Create 'Items' container. To see your container, go to Data Explorer and find the ToDoList database.
- Step 2: Download and run your .NET app**

Once container is created, download a sample .NET app connected to it, extract, build and run.

Download

The screenshot shows the Microsoft Azure portal interface, specifically the 'Data Explorer' section for the 'dxccosmosdb1' account. The left sidebar is identical to the previous screenshot. The main content area is titled 'dxccosmosdb1 | Data Explorer' and displays the following interface:

- SQL API** (highlighted with a yellow box) is selected.
- DATA** (highlighted with a yellow box) is selected under the SQL API.
- NOTEBOOKS** is listed under the SQL API.
- A message states: "Notebooks is currently not available. We are working on it."
- Welcome to Cosmos DB**: "Globally distributed, multi-model database service for any scale."
- Launch quick start**: "Launch a quick start tutorial to get started with sample data"
- New Container**: "Create a new container for storage and throughput"
- Connect**: "Prefer using your own choice of tooling? Find the connection string you need to connect"
- Recents**: A list of recent items.
- Top 3 things you need to know**:
 - Advanced Modeling Patterns**: "Learn advanced strategies to optimize your database."
 - Partitioning Best Practices**: "Learn to apply data model and partitioning strategies."
 - Plan Your Resource Requirements**: "Get to know the different configuration choices."
- Learning Resources**:
 - Get Started using an SDK**: "Learn about the Azure Cosmos DB SDK."
 - Master Complex Queries**: "Learn how to author complex queries."
 - Migrate Your Data**: "Migrate data using Azure services and open-source solutions."

Welcome to CosmosDB

Globally distributed, multi-model database service

New Container

* Database id Create new Use existing

* Container id Container Partition key

For small workloads, the item ID is a suitable choice for the partition key.

+ Add unique key

Analytical store On Off

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. [Learn more](#)

Enable

Recents

Top 3 things you need to know

- [Launch quick start](#) Launch a quick start tutorial to get started with sample data
- [New Container](#) Create a new container for storage and throughput
- [Advanced Modeling Patterns](#) Learn advanced strategies to optimize your database.

[Partitioning Best Practices](#) Learn to apply data model and partitioning strategies.

[Plan Your Resource Requirements](#) Get to know the different configuration choices.

Welcome to CosmosDB

Globally distributed, multi-model database service

Data Explorer

DATA

Music Musicians

- Items
- Settings
- Stored Procedures
- User Defined Functions
- Triggers

NOTEBOOKS

Notebooks is currently not available. We are working on it.

Recents

Top 3 things you need to know

- [Launch quick start](#) Launch a quick start tutorial to get started with sample data
- [New Container](#) Create a new container for storage and throughput
- [Advanced Modeling Patterns](#) Learn advanced strategies to optimize your database.

[Partitioning Best Practices](#) Learn to apply data model and partitioning strategies.

Microsoft Azure | dxccosmosdb1 - Microsoft Azure

https://portal.azure.com/#@ajaykumarbk75gmail.onmicrosoft.com/resource/subscriptions/11b702f9-a331-45f3-acb0-9c7626b0dfa... A Save Discard Upload Item

Search resources, services, and docs (G+)

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxccosmosdb1 | Data Explorer ...

Azure Cosmos DB account

Search (Ctrl+/)

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Quick start Notifications Data Explorer

Settings Features Default consistency Backup & Restore Firewall and virtual networks Private Endpoint Connections CORS Dedicated Gateway

SQL API DATA Music Musicians Items Settings Stored Procedures User Defined Functions Triggers NOTEBOOKS Notebooks is currently not available. We are working on it.

Musicians - It... SELECT * FROM c Edit Filter

id /Albums

1 {
2 "id": "replace_with_new_document_id"
3 }
4
5 }

Microsoft Azure | dxccosmosdb1 - Microsoft Azure

https://portal.azure.com/#@ajaykumarbk75gmail.onmicrosoft.com/resource/subscriptions/11b702f9-a331-45f3-acb0-9c7626b0dfa... A Save Discard Upload Item

Search resources, services, and docs (G+)

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxccosmosdb1 | Data Explorer ...

Azure Cosmos DB account

Search (Ctrl+/)

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Quick start Notifications Data Explorer

Settings Features Default consistency Backup & Restore Firewall and virtual networks Private Endpoint Connections CORS Dedicated Gateway Keys

SQL API DATA Music Musicians Items Settings Stored Procedures User Defined Functions Triggers NOTEBOOKS Notebooks is currently not available. We are working on it.

Musicians - It... SELECT * FROM c Edit Filter

id /Albums

1 {
2 "01": "Michael Jackson",
3 "02": "ABRahman",
4 "03": "KK",
5 "04": "BillieJeans",
6 "05": "JimMorrison"
7 }
8
9
10
11
12 }

Key Value

pythont_azure203.ipynb - Colab | dxccosmosdb1 - Microsoft Azure

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxccosmosdb1 | Data Explorer

Azure Cosmos DB account

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Quick start | Notifications | Data Explorer | Settings | Features | Default consistency | Backup & Restore | Firewall and virtual networks | Private Endpoint Connections | CORS | Dedicated Gateway

SQL API | DATA | Music | Musicians | Items | id | /Albums | Edit Filter

Musicians - It... | SELECT * FROM c | Load more | New Item | Update | Discard | Delete | Upload Item | Edit Filter

Items | id | /Albums | a3965f71-e3...

Settings | Stored Procedures | User Defined Functions | Triggers

NOTEBOOKS | Notebooks is currently not available. We are working on it.

```
1: { "id": "a3965f71-e3...", "name": "Michael Jackson", "age": 50, "genre": "KX", "albums": [ "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f", "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f" ], "self": "dbs/krykyA0JLBQc/colls/krykyA0JLBQc/docs/krykyA0JLBQcBAAAAAAA==/", "etag": "\\"00009002-0000-0100-0000-629842670000\\\"", "attachments": "attachments/", "ts": 1654145639 }
```

pythont_azure203.ipynb - Colab | dxccosmosdb1 - Microsoft Azure

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxccosmosdb1 | Data Explorer

Azure Cosmos DB account

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Quick start | Notifications | Data Explorer | Settings | Features | Default consistency | Backup & Restore | Firewall and virtual networks | Private Endpoint Connections

SQL API | DATA | Music | Musicians | Items | id | /Albums | Edit Filter

Musicians - It... | SELECT * FROM c | Load more | New Item | Update | Discard | Delete | Upload Item | Edit Filter

Items | id | /Albums | 9bfe61f9-00f...

Settings | Stored Procedures | User Defined Functions | Triggers

NOTEBOOKS | Notebooks is currently not available. We are working on it.

```
1: { "id": "9bfe61f9-00f...", "name": "Elvis Presley", "age": 77, "genre": "KX", "albums": [ "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f", "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f" ], "self": "dbs/krykyA0JLBQc/colls/krykyA0JLBQc/docs/krykyA0JLBQcBAAAAAAA==/", "etag": "\\"00009002-0000-0100-0000-62984380000\\\"", "attachments": "attachments/", "ts": 1654145934 }
```

pythont_azure203.ipynb - Colab | dxccosmosdb1 - Microsoft Azure

Microsoft Azure | Search resources, services, and docs (G+)

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxccosmosdb1 | Data Explorer

Azure Cosmos DB account

Overview | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Quick start | Notifications | Data Explorer | Settings | Features | Default consistency | Backup & Restore | Firewall and virtual networks | Private Endpoint Connections

SQL API | DATA | Music | Musicians | Items | id | /Albums | Edit Filter

Musicians - It... | SELECT * FROM c | Load more | New Item | Update | Discard | Delete | Upload Item | Edit Filter

Items | id | /Albums | a3965f71-e3...

Settings | Stored Procedures | User Defined Functions | Triggers

NOTEBOOKS | Notebooks is currently not available. We are working on it.

```
1: { "id": "a3965f71-e3...", "name": "Whilmen Joshi", "age": 32, "genre": "KX", "albums": [ "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f", "a3728cc4-c5f5-4fb9-9888-0f1c57f5e5f" ], "self": "dbs/krykyA0JLBQc/colls/krykyA0JLBQc/docs/krykyA0JLBQcBAAAAAAA==/", "etag": "\\"00009002-0000-0100-0000-62984480000\\\"", "attachments": "attachments/", "ts": 1654146058 }
```

pyhton_azure203.ipynb - Colab x dxccosmosdb1 - Microsoft Azure x +

https://portal.azure.com/#@ajaykumarbk7@gmail.onmicrosoft.com/resource/subscriptions/11b702f9-a331-45f3-acb0-9c7626b0dfa... Añadir Buscar Favoritos

Microsoft Azure

Home > Microsoft.Azure.CosmosDB-20220602095741 > dxccosmosdb1

dxcosmosdb1 | Data Explorer ...

Azure Cosmos DB account

Musicians - It... Query 1 x

Execute Query Save Query

SQL API

Results Query Stats

1 - 3

```
1  SELECT * FROM c
```

{
 "0": {"name": "Michael Jackson",
 "02": "ARRahman",
 "03": "KK",
 "04": "Billie jeans",
 "05": "Jim Morrison",
 "id": "a3965f71-e365-4f41-a67e-565a8c9ab3f0",
 "rid": "KrkjA0JLBQcBAAAAAAA=AAA=",
 "self": "db:/KrkjA0JLBQc-/colls/KrkjA0JLBQc-/docs/KrkjA0JLBQcBAAAAAAA=AAA=/",
 "etag": "\\"00007102-0000-0100-0000-629842670000\"",
 "attachments": "attachments/",
 "_ts": 1654145639
 },
 {
 "0": "Elvis Presley",
 "02": "Jimi Hendrix",
 "03": "The Beetles John Lennon",
 "04": "George Harrison",
 "id": "9bfe01f9-00f8-4b42-8097-86bbe01dac5f",
 "rid": "KrkjA0JLBQcCAAAAAAAA=AAA=",
 "self": "db:/KrkjA0JLBQc-/colls/KrkjA0JLBQc-/docs/KrkjA0JLBQcCAAAAAAAA=AAA=/",
 "etag": "\\"00009a02-0000-0100-0000-6298438e0000\"",
 "attachments": "attachments/",
 "_ts": 1654145934
 },
}

(Django-CRM-master) - Sublime Text (UNREGISTERED)

File Edit Selection Find Goto Tools Project Preferences Help

FOLDERS

Django-CRM-master

- accounts
- cases
- common
- contacts
- crm
- docs
- emails
- events
- invoices
- leads
- marketing
- opportunity
- planner
- static
- tasks
- teams
- templates
- coveragerc
- gitignore
- gitlab-ci.yml
- travis.yml
- db.env
- docker-compose.yml
- Dockerfile
- entrypoint.sh
- LICENSE
- manage.py
- MANIFEST.in
- pytest.ini
- README.rst
- requirements.txt
- setup.py
- system_requirements.txt
- wait-for-postgres.sh

dbenv x | _init_.py x | untitled

```
1 {  
2     "id": "AndersenFamily",  
3     "lastName": "Andersen",  
4     "address": {  
5         "state": "WA",  
6         "county": "King",  
7         "city": "Seattle"  
8     },  
9     "creationDate": 1431620472,  
10    "isRegistered": true  
11 }
```

Line 11, Column 2 Spaces: 2 JSON

Three screenshots of the Microsoft Azure Data Explorer interface showing different JSON documents.

Screenshot 1: Shows a JSON document for the "AndersenFamily" item. The "id" field is highlighted in yellow. A red arrow points from the "id" field in the JSON to the "id" column in the table view below. The table view shows the same document with the "id" field highlighted in yellow.

```

1 {
2   "id": "AndersenFamily",
3   "lastName": "Andersen",
4   "address": {
5     "state": "WA",
6     "county": "King",
7     "city": "Seattle"
8   },
9   "creationDate": 1431620472,
10  "isRegistered": true
11 }

```

id	/Albums
a3965f71-e...	
9bfe619-0...	
a3728cc4-c...	
AndersenFa...	

Screenshot 2: Shows a JSON document for the "WakefieldFamily" item. The "id" field is highlighted in yellow. A red arrow points from the "id" field in the JSON to the "id" column in the table view below. The table view shows the same document with the "id" field highlighted in yellow.

```

1 {
2   "id": "WakefieldFamily",
3   "children": [
4     {
5       "familyName": "Merrim",
6       "givenName": "Jesse",
7       "gender": "female",
8       "grade": 1
9     },
10    {
11      "familyName": "Miller",
12      "givenName": "Lisa",
13      "gender": "female",
14      "grade": 8
15    }
16  }

```

id	/Albums
a3965f71-e...	
9bfe619-0...	
a3728cc4-c...	
AndersenFa...	

Screenshot 3: Shows a JSON document for the "Contoso-C" item. The "id" field is highlighted in yellow. A red arrow points from the "id" field in the JSON to the "id" column in the table view below. The table view shows the same document with the "id" field highlighted in yellow.

```

1 {
2   "id": "Contoso-Checking-Account-2020",
3   "balance": [
4     {
5       "checkingAccount": 1000,
6       "savingsAccount": 5000
7     },
8     {
9       "checkingAccount": 100,
10      "savingsAccount": 5000
11    },
12    {
13      "checkingAccount": -10,
14      "savingsAccount": 5000
15    },
16    {
17      "checkingAccount": 5000,
18      "savingsAccount": 5000
19    }
20  ],
21  "_rid": "KrkYAOJLBQcGAAAAAAA==",
22  "_self": "dbs/KrkYAA=/colls/KrkYAOJLBQc/docs/KrkYAOJLBQcGAAAAAAA==/",
23  "_etag": "(\"00009f02-0000-0100-0000-629847a10000\")",
24  "_attachments": "attachments/",
25  "_ts": 1654146977

```

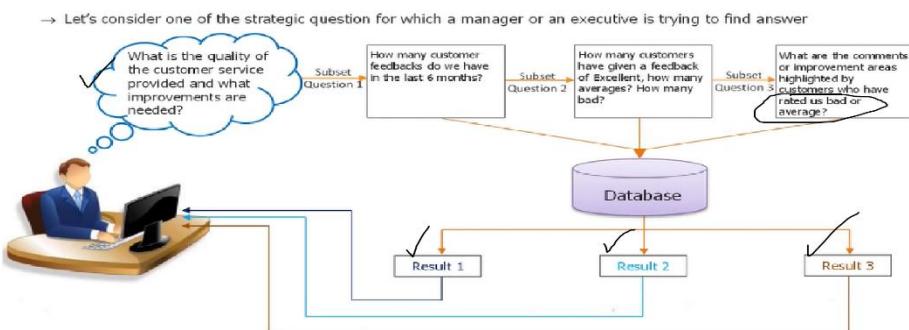
id	/Albums
a3965f71-e...	
9bfe619-0...	
a3728cc4-c...	
AndersenFa...	

QUESTION 5: Explain major difference between databases & data warehouses?

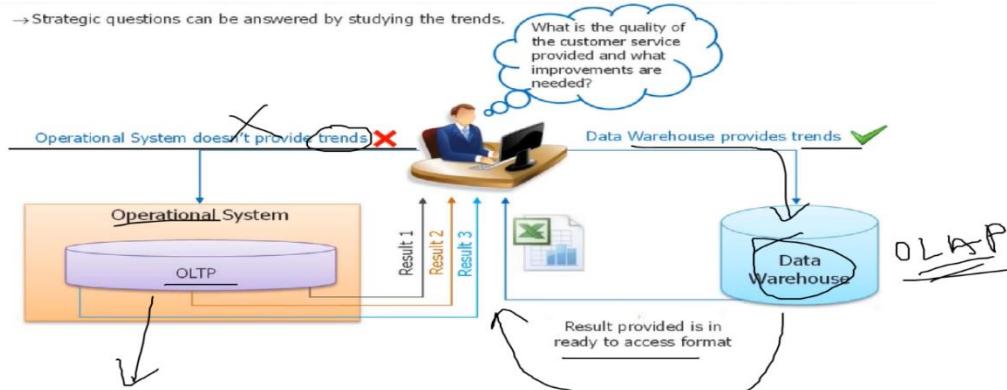
ANSWER:

- The database is designed to capture data, and the data warehouse is designed to analyze data.
- The database is a transaction-oriented design, and the data warehouse is a subject-oriented design.
- The database generally stores business data, and the data warehouse generally stores historical data.
- The database design is to avoid redundancy as much as possible. It is generally designed for a certain business application. For example, a simple User table can record simple data such as user names and passwords. It meets business applications but does not meet analysis. The design of the data warehouse deliberately introduces redundancy. According to the analysis requirements, the analysis dimensions and analysis indicators are designed.

Why do we need Datawarehouse ?



Why do we need Datawarehouse ?



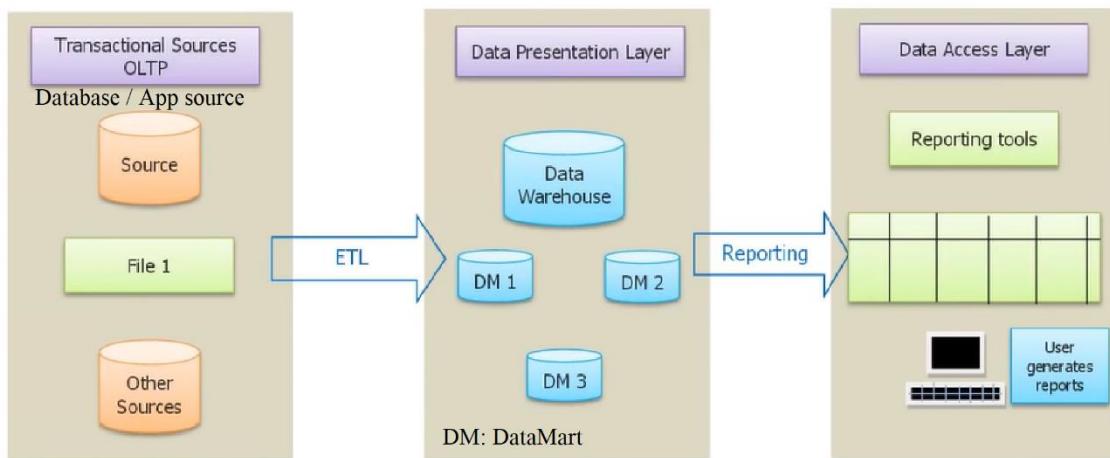
QUESTION 6: Explain the architecture of data warehouses?

ANSWER: Data warehouses and their architectures very depending upon the elements of an organization's situation.

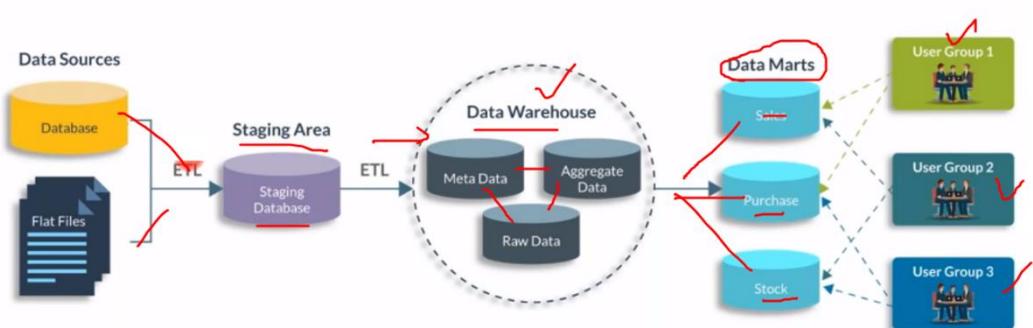
Three common architectures are:

- Data Warehouse Architecture: Basic
- Data Warehouse Architecture: With Staging Area
- Data Warehouse Architecture: With Staging Area and Data Marts

Datawarehouse Architecture



Data Warehouse Architecture



External Sources: External source is a source from where data is collected irrespective of the type of data. Data can be structured, semi structured and unstructured as well.

Staging Area: Since the data, extracted from the external sources does not follow a particular format, so there is a need to validate this data to load into data warehouse. For this purpose, it is recommended to use ETL tool.

- E(Extracted): Data is extracted from External data source.
- T(Transform): Data is transformed into the standard format.
- L(Load): Data is loaded into data warehouse after transforming it into the standard format.

ETL → Extract, Transform & Load

ETL is the process of extracting the data from various sources, transforming this data to meet your requirement and then loading it into a target data warehouse.



After cleansing of data, it is stored in the data warehouse as central repository. It actually stores the meta data and the actual data gets stored in the data marts.

Data mart is also a part of storage component. It stores the information of a particular function of an organisation which is handled by single authority. There can be as many numbers of data marts in an organisation depending upon the functions.

Metadata / Index

- Metadata is defined as data about data.
- Metadata in a DWH defines the source data i.e. Flat File, Relational Database and other objects.
- Metadata is used to define which table is source and target, and which concept is used to build business logic called transformation to the actual output.



QUESTION 7: Explain what are Datamarts & how different from DATABASES?

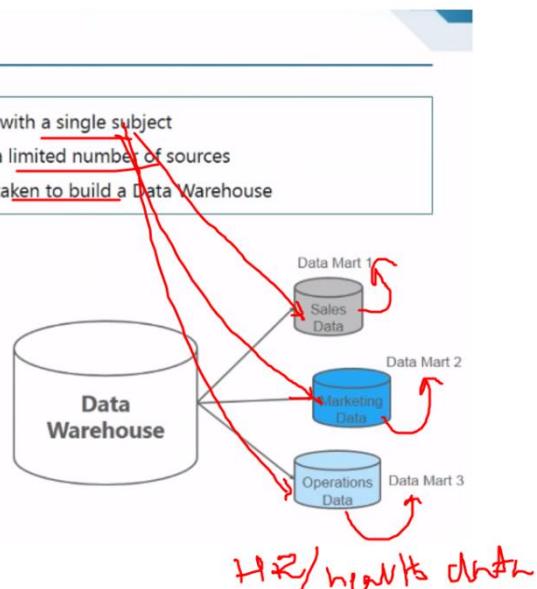
& Mention the types of Datamarts too.

ANSWER: A data mart is a subset of a data warehouse focused on a particular line of business, department, or subject area. Data marts make specific data available to a defined group of users, which allows those users to quickly access critical insights without wasting time searching through an entire data warehouse. For example, many companies may have a data mart that aligns with a specific department in the business, such as finance, sales, or marketing.

Data Mart

- Data mart is a smaller version of the Data Warehouse which deals with a single subject
- Data marts are focused on one area. Hence, they draw data from a limited number of sources
- Time taken to build Data Marts is very less compared to the time taken to build a Data Warehouse

Data Warehouse	Data Marts
Enterprise wide data	Department wide data
Multiple subject areas	Single subject area
Multiple data sources	Limited data sources
Occupies large memory	Occupies limited memory
Longer time to implement	Shorter time to implement



Difference between DataBase and DataMart:

- A database is a transactional data repository (OLTP). A data mart is an analytical data repository (OLAP).
- A database captures all the aspects and activities of one subject in particular. A data mart will house data from multiple subjects.
- The data in a database will be raw and unprocessed (not cleaned). The data in a data mart will be processed and validated for greater reporting ease.
- Users do not interact with data in a database. Users directly interact with data from a data mart.
- Databases are the first step in the data ETL process. Data marts are the last step in the ETL process.

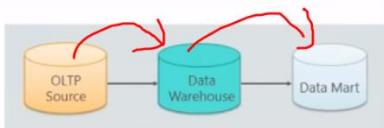
types of Datamarts:

There are three types of data marts: dependent, independent, and hybrid. They are categorized based on their relation to the data warehouse and the data sources that are used to create the system.

Types Of Data Mart

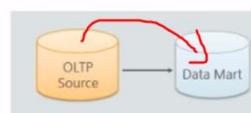
1. Dependent Data Mart

- The data is first extracted from the OLTP systems and then populated in the central DWH
- From the DWH, the data travels to the Data Mart



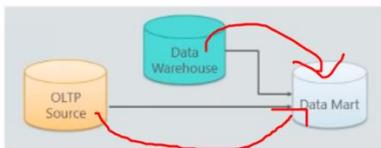
2. Independent Data Mart

- The data is directly received from the source system
- This is suitable for small organizations or smaller groups within an organization



3. Hybrid Data Mart

- The data is fed both from OLTP systems as well as the Data Warehouse



QUESTION 8: Explain OLAP & OLTP with examples?

ANSWER: OLAP stands for On-Line Analytical Processing. It is used for analysis of database information from multiple database systems at one time such as sales analysis and forecasting, market research, budgeting and etc. Data Warehouse is the example of OLAP system.

OLTP stands for On-Line Transactional processing. It is used for maintaining the online transaction and record integrity in multiple access environments. OLTP is a system that manages very large number of short online transactions.

Information Systems:- OLTP (DB) vs. OLAP (DWH)

Relational Database (OLTP) <i>Db</i>	Analytical Data Warehouse (OLAP) <i>DW</i>
Contains current data	Contains historical data
Useful in running the business	Useful in analyzing the business
Based on Entity Relationship Model	Based on Star, Snowflake and Fact Constellation Schema
Provides primitive and highly detailed data	Provides summarized and consolidated data
Used for writing data into the database	Used for reading data from the data warehouse
Database size ranges from 100 MB to 1 GB	Data Warehouse size ranges from 100 GB to 1 TB
Fast; provides high performance	Highly flexible; but not fast
Number of records accessed is in tens	Number of records accessed is in millions
Ex: All bank transactions made by a customer	Ex: Bank transactions made by a customer at a particular time.

student

course

Information Systems:- OLTP (DB) vs. OLAP (DWH)

OLTP Examples:

1. A supermarket server which records every single product purchased at that market.
2. A bank server which records every time a transaction is made for a particular account.
3. A railway reservation server which records the transactions of a passenger.

OLAP Examples:

1. Bank Manager wants to know how many customers are utilizing the ATM of his branch. Based on this he may take a call whether to continue with the ATM or relocate it.
2. An insurance company wants to know the number of policies each agent has sold. This will help in better performance management of agents.

QUESTION 9: Explain what is BI & how BI helps business to take intelligent decisions?

ANSWER: Business intelligence (BI) refers to the procedural and technical infrastructure that collects, stores, and analyzes the data produced by a company's activities.

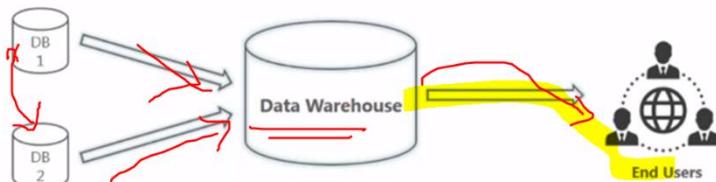
- BI represents the technical infrastructure that collects, stores, and analyzes company data.
- BI parses data and produces reports and information that help managers to make better decisions.
- Software companies produce BI solutions for companies that wish to make better use of their data.
- BI tools and software come in a wide variety of forms such as spreadsheets, reporting/query software, data visualization software, data mining tools, and online analytical processing (OLAP).
- Self-service BI is an approach to analytics that allows individuals without a technical background to access and explore data.

What Is Business Intelligence?

BI is the act of transforming raw/ operational data into useful information for business analysis.

How Does It Work?

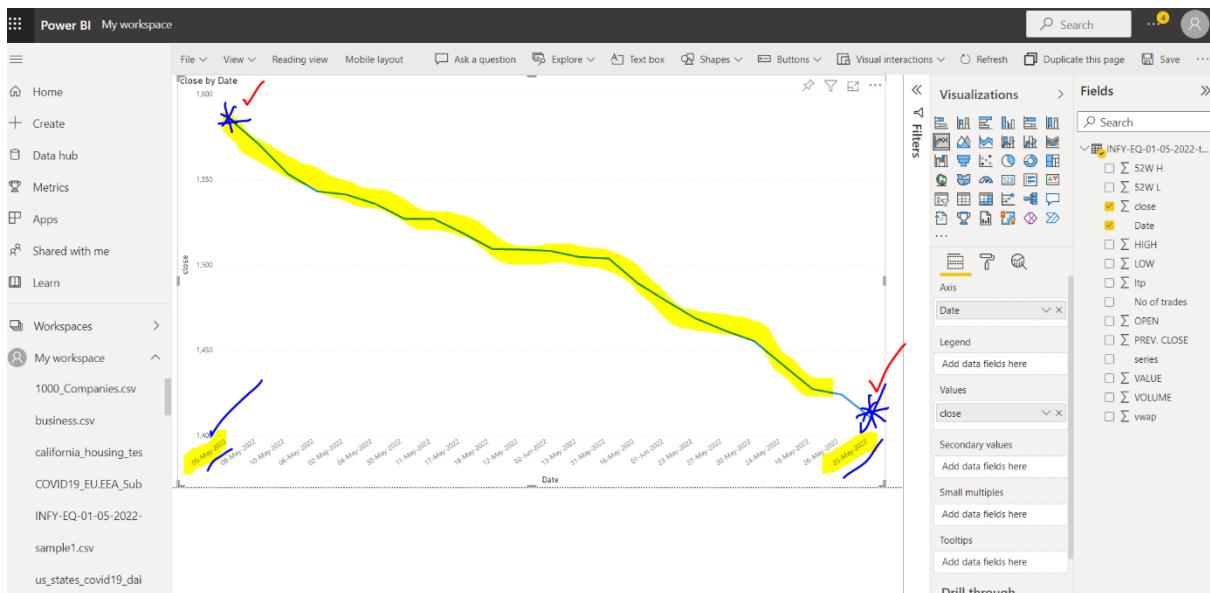
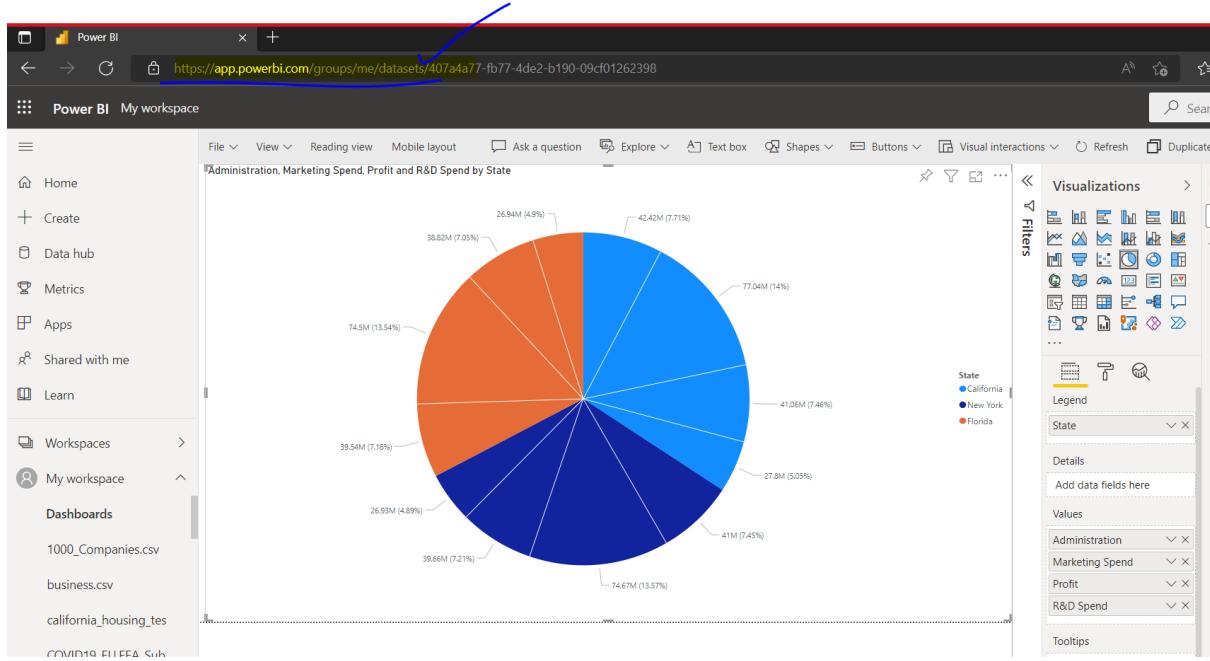
1. BI based on Data Warehouse technology **extracts** information from a company's **operational systems**.
2. The data is **transformed** (**cleaned and integrated**), and **loaded** into Data Warehouses.
3. Since this data is **credible**, it is used for **business insights**.



Why Business Intelligence?

Business Intelligence is the activity which contributes to the growth of any company.

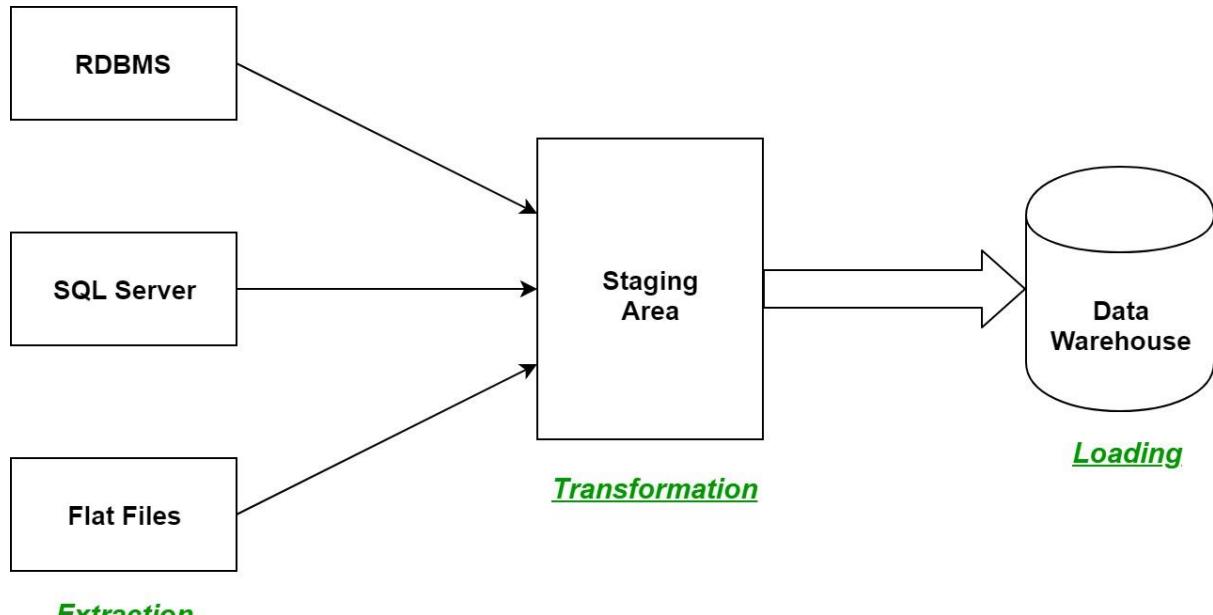




QUESTION 10: Explain how ETL works with Data warehouses?

ANSWER:

ETL is a process in Data Warehousing and it stands for **Extract, Transform and Load**. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.



ETL → Extract, Transform & Load

ETL is the process of extracting the data from various sources, transforming this data to meet your requirement and then loading it into a target data warehouse.



ETL process can also use the pipelining concept i.e. as soon as some data is extracted, it can be transformed and during that period some new data can be extracted. And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed. The block diagram of the pipelining of ETL process is shown below:

THANK YOU!

NAME: ASHOK KUMAR