Q.A 08

Asynchronous Apex

Q) Database class methods to perform Insert, Update, Delete operation in Apex.

Q-1) what is Asynchronous Apex ?

→ (i) An asynchronous process executes a task in the Background.
(ii) User doesn't have to wait for the task to finish.
(iii) Use Asynchronous Apex for :
(i) Callouts to external System.
(ii) Operation that require higher limits.
(iii) Code that needs to run at a certain time.

Q-2) Benefits of Asynchronous processing ?

→ (i) Use efficiency
(ii) Scalability
(iii) Higher limits

Q-3) Types of Asynchronous Processing ?

→

	Type	Overview	Common Scenarios
(i)	Future methods	Run in their own thread, and do not start until resource are available.	web Service Callout.

(2) Batch Apex	Run large jobs that would exceed normal processing limits	Data cleansing or archiving of records
(3) Queueable Apex	Similar to future methods, but provide additional job chaining & allow more complex data types to be used	performing sequential processing operations with external web services.
(4) Scheduled Apex:	Scheduled Apex to run at a specified time	Daily or weekly tasks

Q-4) Governor & Execution limits in Asynchronous processing ?

→ (i) Asynchronous apex provides higher governor and execution limits.

(ii) Number of SOQL is doubled from 100 to 200.

(iii) Total heap size and maximum CPU time are similarly larger for asynchronous calls.

(iv) As you get higher limits with async also those governor limits are independent of the limits in the synchronous request that queued the async request initially

Q-5) Challenges of Asynchronous processing?

→ (i) Ensure fairness of processing.
(ii) Ensure fault tolerance.

Q-6) what is Future Apex / method?

→ (i) Future Apex runs process in a Separate thread, at a later time when system resources become available.
(ii) Use @future annotation to create future methods.
(iii) In Synchronous processing, all method calls are made from the same thread and no additional processing can occur until the process is complete.
(iv) whereas in future method, methods run asynchronously in it's own thread.
(v) This unblocks users from performing other operations.
(vi) provides higher governor & execution limits for processing.

Q-7) when to use future methods?

→ (i) Callouts to external web Services. To make callouts from a trigger use a future or queuable method.
(ii) Process that needs to executed in a Separate on this own thread.
(iii) Isolating DML operations on different sObject types to prevent the mixed DML error.

Q-8) Best practice : Future Method &

→ (1) Ensure future method execute as fast as possible

(2) In case of web service callouts, try to bundle all callouts together from the same future method, rather that using a seperate future method for each callout.

(3) Test that a trigger enqueuing the @future calls is able to handle a trigger collection of 200 records.

(4) To process large number of records asynchronously, use Batch Apex instead of future methods.

Q-9) Things to Remember while using future method

→ (i) @future annotation method must be static.

(ii) Future method can only return a void

(iii) Future method can take primitive data types, array of primitive data types, or collections of primitive data types as arguments.

(iv) Future method cannot take objects as argu

(v) It can happen that future method from a future method. Nor can you invoke a trigger that calls a future method while running a future method.

(vi) There is a limit of 50 future calls per apex invocation.

(vii) There is an additional limit on the number of calls in a 24-hour period.

Q.10) What is Batch Apex ?

→ (i) Batch Apex runs / carries jobs. it can process thousands or millions of record
(ii) it processes records asynchronously in batches.
(iii) For Data cleansing or archiving Batch Apex is probably best solution.

Q.11) How Batch Apex works ?

→ (i) The execution logic of the batch class is called once for each batch of records that is being processed.
(ii) Each time when a batch class is invoked This job is placed on the Apex job queue and is executed as a discrete transaction.

Advantages are :

(1) Every transaction start with a new set of governor limits.
(2) if one batch fails to process Successfully all other Successful batch transactions aren't rolled back.

Q.12) ~~what~~ which methods are used in Batch Apex ?

→ (1) Batch Apex class must implement the Database.Batchable interface and include the following three methods :
(i) Start
(2) execute
(3) finish

Q-13) what is the use of start() method in Batch Apex ?

→ (i) Collects the records or objects to be passed to the execute() method for processing.

(ii) start() is called once at the beginning of a Batch Apex Job.

(iii) it returns a Database.QueryLocator object or an Iterable that contains the records or objects passed into the job

(iv) when QueryLocator object is used, the governor limit for the total number of records retrieved by SOQL queries is bypassed and upto 50 million records can be queried.

(v) whereas with an Iterable governor limit by SOQL query is enforced.

Q-14) what is the use of execute() method in Batch Apex ?

→ (i) Perform actual processing for each batch of data passed.

(ii) Default batch size is 200 records.

(iii) Batches of records can execute in any order. it doesn't depends on which order they are received from the start method.

(iv) it take a reference to the Database.BatchableContext object and A list<sObject> or a list of parameterized types.

⑧ when using Database.QueryLocator use the returned list.

Q.15) what is the use of finish() method in Apex ?

→ ① Executes post-processing operations.
② Calls once after all batches are processed.
③ For example, Sending an email process can be implemented in finish method. job.

Q.16) Batch practices: Batch Apex ?

→ ① To ensures fast execution of batch jobs, minimize web service callout time.
② Tune any SOQL query to gather the records to execute as quickly as possible
③ The longer the batch job executes, the more likely other queued jobs are delayed when many jobs are in the queue.
④ Use Batch Apex when more than one batch of records are there, in case of one batch, you can prefer Queueable apex.
⑤ Minimize the number of asynchronous request.
⑥ If you are planning to invoke a batch job from a trigger then you must be able to guarantee that the trigger won't add more batch jobs than the limit.

Q-17) what is Queueable Apex ?

→ (1) Superset of future methods with extra features.
(2) Combination of future methods and Batch Apex.
(3) Works beyond primitive Arguments.
(4) called by a simple System.enqueueJob() method
(5) enqueueJob() return a job ID that can be monitored.

Q-18) what are the benefits of Queueable Apex ?

→ (1) Non-primitive types
(2) Monitoring
(3) chaining Jobs.

Q-19) Important to Remember while using Queueable Apex :

→ (1) The execution of a queued job counts once against the shared limit for asynchronous Apex method executions.
(2) you can add up to 50 Jobs to the queue with System.enqueueJob() in a single transaction.
(3) when ~~chaining~~ chaining jobs, you can add only one job from an executing job with System.enqueueJob().
(4) No limit is enforced on the depth of chained jobs. Note for Developer Edition and Trial Orgs, the maximum stack depth for chained jobs is 5 (including the initial parent queueable job).

Q-20) →

Q-21)

Q-22) →

Q-20) What is scheduled Apex ?

→ (1) You can run Apex classes at a specified time.
(2) Run maintenance tasks on Daily as well as weekly basis.
(3) Implements Schedulable interface in an Apex class method.

Q-21) How many ways are there to schedule Apex ?

→ (1) Using the System.Schedule() method through CRON expression.
① System.schedule(JobName, CronExpression, Schedulable<ClassInstance>);
(2) Scheduling a job from the UI.
① Setup → Apex Classes → Schedule Apex

Q-22) What is CRON Expression ?

→ ① A CRON expression is basically a string of five or six fields that represents a set of time, normally as a schedule to execute some routine.

Example :
① String sch = '20 30 8 10 26 2022';
② 20 = Seconds
③ 30 = Minutes
④ 8 = Hours
⑤ 10 = Days of month
⑥ 2 = month (1 for Jan and 12 for Dec)
⑦ ? = Day of week (1 for Sun and 7 for Sat)
⑧ 2022 = Year (Optional - null or 1970 - 2099)

Q-23) Things to Remember while Scheduling Apex

→ ① You can only have 100 scheduled jobs at one time.

② while scheduling a class from a trigger then you must guarantee that the trigger won't add more scheduled jobs than the limit.

③ Synchronous web services callouts are not supported from scheduled Apex

④ make an Asynchronous callout by placing the callout in a method annotated with @future(callout=true) and call this method from scheduled Apex

⑤ If scheduled apex executes a batch job then callouts are supported from the batch class