Question 01 :

-- QUESTION 1

- Create a view that displays basic user information including

- their display name, reputation, location, and account creation date.

- Name the view: vw_BasicUserInfo

- Test the view by selecting all records from it.

Question 02 :

Create a view that shows all posts with their titles, scores,

- view counts, and creation dates where the score is greater than 10.

- Name the view: vw_HighScoringPosts

- Test by querying posts from this view.

Question 03 :

Create a view that combines data from Users and Posts tables.

- Show the post title, post score, author name, and author reputation.

- Name the view: vw_PostsWithAuthors

- This is a complex view involving joins.

Question 04 :

Create a view that aggregates comment statistics per post.

- Include: PostId, total comment count, sum of comment scores,

- and average comment score.

- Name the view: vw_PostCommentStats

- This is a complex view with aggregation.

Question 05 :

Create an indexed view that shows user activity summaries.

- Include: UserId, DisplayName, Reputation, total posts count.

- Name the view: vw_UserActivityIndexed

- Make it an indexed view with a unique clustered index on UserId

Question 06 :

 Create a partitioned view that combines high reputation users

- (reputation > 5000) and low reputation users (reputation <= 5000)

- from the same Users table using UNION ALL.

- Name the view: vw_UsersPartitioned

.

Question 07 :

 Create an updatable view on the Users table that shows

- UserId, DisplayName, and Location.

- Test the view by updating a user's location through the view.
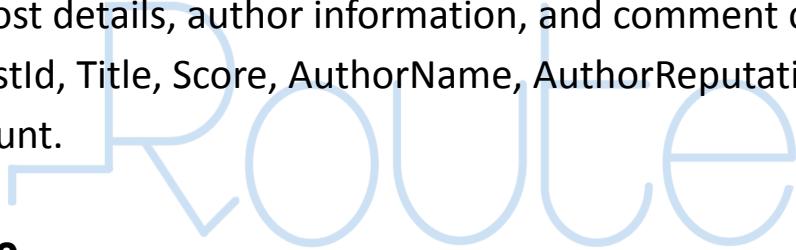
- Name the view: vw_EditableUsers

.

Question 08 :

Create a view with CHECK OPTION that only shows posts with
- score greater than or equal to 20.
- Name the view: vw_QualityPosts
- Ensure that any updates through this view maintain the score >= 20 condition .

Question 09 :

 Create a complex view that shows comprehensive post information
- including post details, author information, and comment count.
- Include: PostId, Title, Score, AuthorName, AuthorReputation, CommentCount.

Question 10 :

Create a view that shows badge statistics per user.
- Include: UserId, DisplayName, Reputation, total badge count,
- and a list of unique badge names (comma-separated if possible,
- or just the count for simplicity).
- Name the view: vw_UserBadgeStats .

Question 11 :

Create a view that shows only active users (those who have
- posted in the last 365 days from today, or have a reputation > 1000).
- Include: UserId, DisplayName, Reputation, LastActivityDate
- Name the view: vw_ActiveUsers.

Question 12 :

Create an indexed view that calculates total views and average

- score per user from their posts.

- Include: UserId, TotalPosts, TotalViews, AvgScore

- Name the view: vw_UserPostMetrics

- Create a unique clustered index on UserId.

Question 13 :

Create a view that categorizes posts based on their score ranges.

- Categories: 'Excellent' (>= 100), 'Good' (50-99), 'Average' (10-49),

  'Low' (< 10)

- Include: PostId, Title, Score, Category

- Name the view: vw_PostsByCategory