

Writing Prompts in BARD

General steps to follow:

- Do not rely on BARD results. Always validate your results with traditional methods of MySQL.
- The results provided by BARD may vary slightly with multiple execution of same prompts.
- Open the link '<https://bard.google.com/>' and login with your Gmail credentials.
- A chat window is visible to write the prompts/queries.
- 'Write' or 'copy and paste' the exact same prompts in chat window of BARD. If you choose to 'copy and paste' a query, right-click on the BARD chat window and select 'Paste as plain text' or press 'Ctrl+Shift+V' for optimal formatting.

Demo 7.1: User Responses by BARD

Prompt given to BARD:

1. You're a local guide giving me travel advice for Italy. Construct a 2-day itinerary highlighting the city's most popular places, including stops with a skyline view, a museum, and a historical landmark.

Prompt given to BARD:

2. I'm interviewing for a Data Analyst position and want to be well-prepared. Assume the role of an expert interviewer in this field and provide me with a list of questions that are likely to be asked during the interview.

Demo 7.2: Motion Pictures Data Analysis using BARD

1. As a MySQL expert, consider the following structure of table. Write the MySQL query to extract the names of all the actors in their movie collection. Display the first names, last names, actor IDs, and the last updated details.

Actor(actor_id smallint unsigned, first_name varchar(45), last_name varchar(45), last_update timestamp)

2. Write the MySQL query to calculate and display the average rental rates in descending order based on the movie ratings.

3. As a MySQL expert, consider the following structures of tables. Write the MySQL query, provide the list of movies along with the number of actors listed for each movie.
 - a. Film(film_id smallint unsigned, title varchar(128), description text, release_year year, language_id tinyint unsigned, original_language_id tinyint unsigned, rental_duration tinyint unsigned, rental_rate decimal(4,2), length smallint unsigned, replacement_cost decimal(5,2), rating enum('G','PG','PG-13','R','NC-17'), special_features set('Trailers','Commentaries','Deleted Scenes','Behind the Scenes'), last_update timestamp)
 - b. Film_Actor(actor_id smallint unsigned, film_id smallint unsigned, last_update timestamp)

4. The movie 'AGENT TRUMAN' has been a great success. Write the MySQL query to display the first names and last names of all actors who are a part of this movie.

5. As a MySQL expert, consider the following structures of 3 tables. Consider Sales has been down among the family audience with kids. Write the MySQL query to display the movies that fall under the 'children' category. Also, write the query to identify and display the names of the movies in the family category.
 - a. Film(film_id smallint unsigned, title varchar(128), description text, release_year year, language_id tinyint unsigned, original_language_id tinyint unsigned, rental_duration tinyint unsigned, rental_rate decimal(4,2), length smallint unsigned, replacement_cost decimal(5,2), rating enum('G','PG','PG-13','R','NC-17'), special_features set('Trailers','Commentaries','Deleted Scenes','Behind the Scenes'), last_update timestamp)
 - b. Category(category_id tinyint unsigned, name varchar(25), last_update timestamp)
 - c. Film_category(film_id smallint unsigned, category_id tinyint unsigned, last_update timestamp)

6. As a MySQL expert, consider the following structures of 3 tables. Write a query to display category wise category names, average rental_rate as 'Avg_rental_price', average replacement_cost as 'Avg_replacement_price', difference of average replacement cost to average rental rate as "replace_sub_rental" with the condition difference of average replacement cost to average rental rate is greater than 15.
 - a. Film(film_id smallint unsigned, title varchar(128), description text, release_year year, language_id tinyint unsigned, original_language_id tinyint unsigned, rental_duration tinyint unsigned, rental_rate decimal(4,2), length smallint unsigned, replacement_cost decimal(5,2), rating enum('G','PG','PG-13','R','NC-

17'),special_features set('Trailers','Commentaries','Deleted Scenes','Behind the Scenes'), last_update timestamp)

- b. Category(category_id tinyint unsigned, name varchar(25), last_update timestamp)
- c. Film_category(film_id smallint unsigned, category_id tinyint unsigned, last_update timestamp)