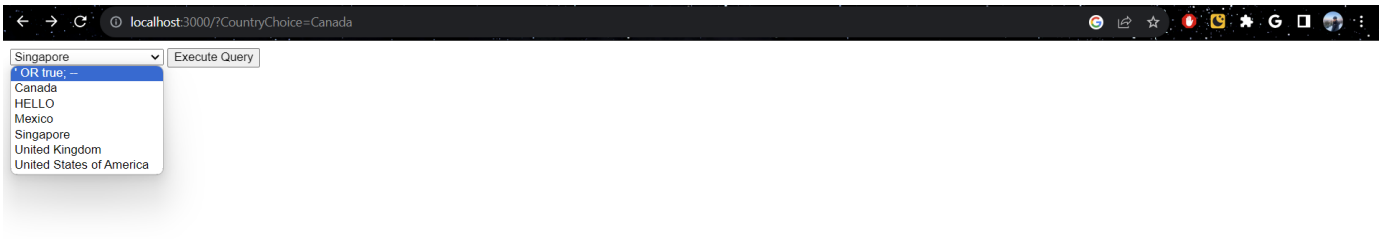
1.

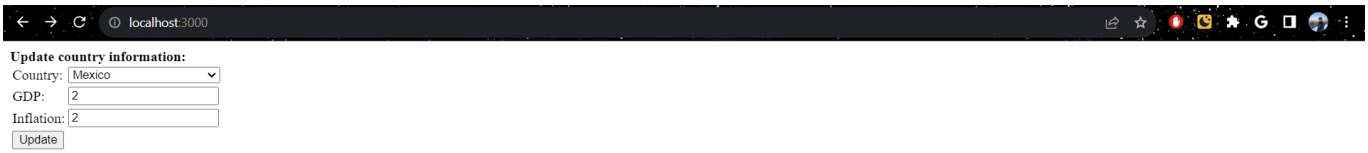

localhost:3000/?PetTypeChoice
hw-6.pdf
localhost:3000/?PetTypeChoice=bird
bird
Execute Query
bird
cat
chicken
dog
duck
fish
pig

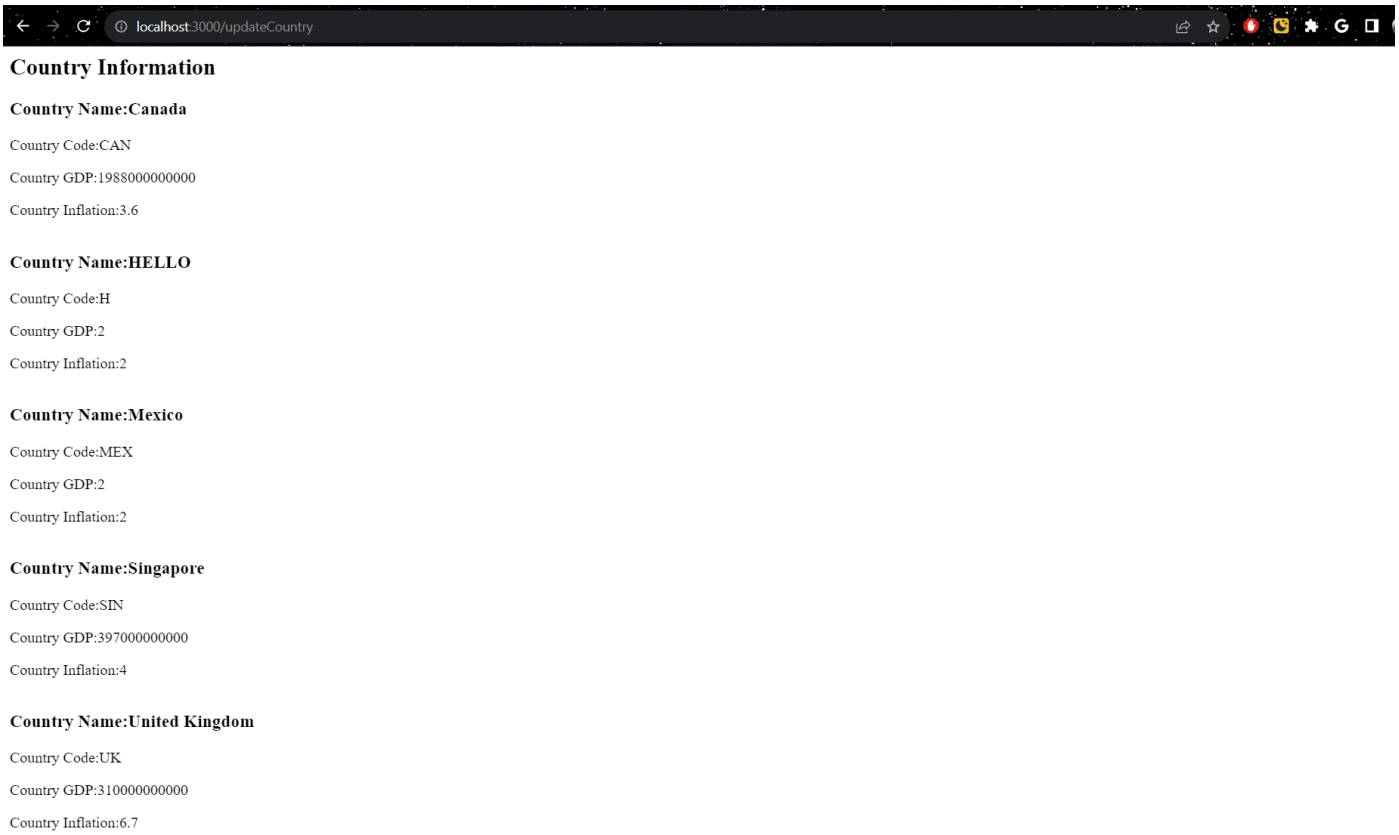
2.


localhost:3000/?CountryChoice=Canada
Singapore
Execute Query
' OR true; --
Canada
HELLO
Mexico
Singapore
United Kingdom
United States of America


localhost:3000/country
Country Information
Country Name: Singapore
Country Code: SIN
GDP: 397000000000
Inflation: 4

3.


localhost:3000
Update country information:
Country: Mexico
GDP: 2
Inflation: 2
Update


localhost:3000/updateCountry
Country Information
Country Name:Canada
Country Code:CAN
Country GDP:1988000000000
Country Inflation:3.6
Country Name:HELLO
Country Code:H
Country GDP:2
Country Inflation:2
Country Name:Mexico
Country Code:MEX
Country GDP:2
Country Inflation:2
Country Name:Singapore
Country Code:SIN
Country GDP:397000000000
Country Inflation:4
Country Name:United Kingdom
Country Code:UK
Country GDP:310000000000
Country Inflation:6.7

4. Books(<u>BookID</u>, Title, Author, ISBN)
   (a) Functional Dependencies:
      (i) Primary Key → BookID
      (ii) Non-trivial dependency: Author → Title
      (iii) Non-trivial dependency: ISBN → Title, Author
   (b) Redundant Data: This instance of the relation shows redundancy as two books have the exact same ISBN

<u>BookID</u>	**Title**	**Author**	**ISBN**
1	Intro to Databases	John Smith	123-456-789
2	Database Systems	Jane Doe	123-456-789

   (c) Anomalies:
      (i) Update Anomaly: Say we need to update the author's name for a specific book. In the instance above where two books share the same ISBN, we would need to update multiple rows since Author is dependent on the ISBN.
      (ii) Insertion Anomaly: Say we wanted to add a new book with a new ISBN but we don't have information about the author. We cannot do this as Author is dependent on ISBN.
      (iii) Deletion Anomaly: If we were to delete a row for a book with specific ISBN, we could unintentionally remove information from books with the same ISBN.

5.
   (a) X → Y and YW → Z, then XW → Z
      (i) Use Augmentation rule: If Y ⊆ X, then XW → YW
      (ii) Use Transitivity: If XW → YW and YW → Z, then XW → Z
      (iii) So, If X → Y and YW → YZ, then XW → Z
   (b) If X → YZ, then X → Y and X → Z
      (i) Use Augmentation rule: If Z ⊆ YZ, then X → YZ implies XZ → YZ
      (ii) Use Reflexivity rule: If YZ ⊆ YZ, then XZ → implies X → Z
      (iii) Use Augmentation rule: If Y ⊆ YZ, then X → Z implies XY → Z
      (iv) So, If X → YZ, then X → Y and X → Z
   (c) If X → Y and X → Z, then X → YZ
      (i) Use Augmentation rule: If Z ⊆ Y, then X → Y implies XZ → YZ
      (ii) Use Transitivity rule: If X → Y and XZ → YZ, then X → YZ

6.
   (a) R(a, b, c) where a → c
      (i) Candidate key: a
   (b) R(a, b, c, d) where b → c and d → a
      (i) Candidate key: (b, d)
   (c) R(a, b, c, d) where a → c and c → d
      (i) Candidate key: (a, c)
   (d) R(a, b, c, d, e)
      (i) Candidate key: (c, b, d)

7.

1. album relation:

(a) Relational Schema:
- album(title, year_recorded, record_label, group_id)

(b) Non-key, non-trivial functional dependencies:
- title -> year_recorded
- year_recorded -> title
- title, record_label -> group_id
- group_id -> title

(c) Corresponding Normal Form:
- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

2. music_group relation:

(a) Relational Schema:
- music_group(group_id, group_name, year_created)

(b) Non-key, non-trivial functional dependencies:
- group_id -> group_name
- group_name -> group_id
- year_created -> group_id

(c) Corresponding Normal Form:
- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

3. genre relation:

(a) Relational Schema:
- genre(genre_label, genre_descrip)

(b) Non-key, non-trivial functional dependencies:
- genre_label -> genre_descrip
- genre_descrip -> genre_label

(c) Corresponding Normal Form:
- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

4. group_genre relation:

(a) Relational Schema:
- group_genre(group_id, genre_label)

(b) Non-key, non-trivial functional dependencies:
- group_id -> genre_label
- genre_label -> group_id

(c) Corresponding Normal Form:
- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

5. artist relation:

(a) Relational Schema:

- artist(artist_id, artist_name, birth_year)

(b) Non-key, non-trivial functional dependencies:

- artist_id -> artist_name
- artist_name -> artist_id
- birth_year -> artist_id

(c) Corresponding Normal Form:

- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

6. group_membership relation:

(a) Relational Schema:

- group_membership(artist_id, group_id, start_year)

(b) Non-key, non-trivial functional dependencies:

- artist_id, group_id -> start_year
- start_year -> artist_id, group_id

(c) Corresponding Normal Form:

- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

7. track relation:

(a) Relational Schema:

- track(track_id, track_name, year_rec)

(b) Non-key, non-trivial functional dependencies:

- track_id -> track_name
- track_name -> track_id
- year_rec -> track_id

(c) Corresponding Normal Form:

- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

8. song relation:

(a) Relational Schema:

- song(song_title, year_written, artist_id)

(b) Non-key, non-trivial functional dependencies:

- song_title -> year_written
- year_written -> song_title
- artist_id -> song_title

(c) Corresponding Normal Form:

- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.

9. track_artists relation:

(a) Relational Schema:

- track_artists(track_id, artist_id)

(b) Non-key, non-trivial functional dependencies:

- track_id -> artist_id
- artist_id -> track_id

(c) Corresponding Normal Form:

- The relation is in at least the Third Normal Form (3NF) since there are no transitive dependencies or non-prime attributes dependent on part of a candidate key.
-