Chess Extension Testing Unit

December 2023

1 DataTypes

1.1 chessboard Datatype

To check the size of chessboard datatype

```
SELECT text_to_chessboard('rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/
    RNBQKBNR b KQkq e3 0 1');
```

```
postgres=# select text_to_chessboard('rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1');
INFO:  Size of chessboard datatype: 69 bytes
                  text_to_chessboard
-------------------------------------------------------
 rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1
(1 row)
```

Figure 1: Size of chessboard datatype

1.2 chessgame Datatype

To check the size of chessgame datatype

```
SELECT text_to_chessgame
('1.e4 c5 2.Nf3 d6 3.Bb5+ Nd7 4.d4 Nf6 5.Nc3 cxd4 6.Qxd4 e5 7.Qd3
    h6 8.Be3 Be7 9.O-O O-O 10.Rad1 a6');
```

```
postgres=# SELECT text_to_chessgame('1.e4 c5 2.Nf3 d6 3.Bb5+ Nd7 4.d4 Nf6 5.Nc3 cxd4 6.Qxd4 e5 7.Qd3 h6 8.Be3 Be7 9.O-O O-O 10.Rad1 a6');
INFO:  Size of chessgame datatype: 44 bytes
                                text_to_chessgame
-------------------------------------------------------------------------------------------
 1. e4 c5 2. Nf3 d6 3. Bb5+ Nd7 4. d4 Nf6 5. Nc3 cxd4 6. Qxd4 e5 7. Qd3 h6 8. Be3 Be7 9. O-O O-O 10. Rad1 a6
(1 row)
```

Figure 2: Size of chessgame Datatype

2 Functions

2.1 getBoard

1. The initial state of the chessboard

```
    SELECT getBoard(game, 0) as initial_board
    FROM chessgames;
```

```
postgres=# SELECT getBoard(game, 0) as initial_board FROM chessgames limit 5;
                  initial_board
-----------------------------------------------------------
 rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
 rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
 rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
 rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
 rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1
(5 rows)
```

Figure 3: Initial Board State

2. Displaying the initial position in the table and the board states after the first moves by white and black.

```
    SELECT
    id,
    getBoard(game, 0) as initial_position,
    getBoard(game, 1) as after_white_first_move,
    getBoard(game, 2) as after_black_first_move
    FROM chessgames LIMIT 5;
```

```
 id  |                initial_position                |               after_white_first_move               |               after_black_first_move
-----+------------------------------------------------+----------------------------------------------------+----------------------------------------------------
 1861 | rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1 | rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1 | rnbqkbnr/pp1ppppp/8/2p5/4P3/8/PPPP1PPP/RNBQKBNR
w KQkq c6 0 2
 1862 | rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1 | rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1 | rnbqkbnr/pp1ppppp/8/2p5/4P3/8/PPPP1PPP/RNBQKBNR
w KQkq c5 0 2
 1863 | rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1 | rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1 | rnbqkbnr/pppp1ppp/8/4p3/4P3/8/PPPP1PPP/RNBQKBNR
w KQkq e6 0 2
 1864 | rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1 | rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1 | rnbqkbnr/pppp1ppp/4p3/8/4P3/8/PPPP1PPP/RNBQKBNR
w KQkq - 0 2
 1865 | rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1 | rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1 | rnbqkbnr/pp1ppppp/8/2p5/4P3/8/PPPP1PPP/RNBQKBNR
:
```

Figure 4: Initial position, position after white and black first moves in the board

3. After the first 10 half-moves in games, what are the popular board states that start with the King's Pawn opening?

```
    SELECT
    getBoard(game, 10) AS board_state_after_10_moves,
    COUNT(*) AS number_of_games
    FROM chessgames
    WHERE getFirstMoves(game, 1) = '1.e4 '::chessgame
    GROUP BY board_state_after_10_moves
    ORDER BY number_of_games DESC
    LIMIT 5;
```

```
                board_state_after_10_moves                | number_of_games
----------------------------------------------------------+-----------------
 rnbqkb1r/1p2pppp/p2p1n2/8/3NP3/2N5/PPP2PPP/R1BQKB1R w KQkq - 0 6 |             465
 r1bqk2r/1pppbppp/p1n2n2/4p3/B3P3/5N2/PPPP1PPP/RNBQ1RK1 w kq - 4 6 |             129
 rnbqkb1r/pp3ppp/3ppn2/8/3NP3/2N5/PPP2PPP/R1BQKB1R w KQkq - 0 6 |              66
 rnbqkb1r/ppp2ppp/8/3p4/3Pn3/5N2/PPP2PPP/RNBQKB1R w KQkq - 0 6 |              45
 r1bqkbnr/pp3ppp/2npp3/8/3NP3/2N5/PPP2PPP/R1BQKB1R w KQkq - 0 6 |              36
(5 rows)
```

Figure 5: Distribution of Board States in a Popular Opening

2.2 getFirstMoves

1. Displaying common openings by categorizing and counting the number of chess games by their opening sequence

```
    SELECT getFirstMoves(game, 6) as opening_sequence,
    COUNT(*) as number_of_games
    FROM chessgames
    GROUP BY opening_sequence
    ORDER BY number_of_games DESC LIMIT 5;
```

```
      opening_sequence       | number_of_games
-----------------------------+-----------------
 1. e4 c5 2. Nf3 d6 3. d4 cxd4 |             531
 1. d4 Nf6 2. c4 g6 3. Nc3 Bg7 |             255
 1. d4 Nf6 2. c4 e6 3. Nc3 Bb4 |             228
 1. d4 Nf6 2. c4 g6 3. Nc3 d5  |             216
 1. d4 Nf6 2. c4 e6 3. Nf3 b6  |             198
(5 rows)
```

Figure 6: Common Openings

2. What are the most common board states after 8 moves between the Queen's Gambit and the Sicilian Defense?

```
    SELECT
    getFirstMoves(game, 2) AS opening,
    getBoard(game, 8) AS board_state,
    COUNT(*) AS frequency
    FROM chessgames
    WHERE getFirstMoves(game, 2) IN ('1.d4 d5 2.c4 '::chessgame, '
    1.e4 c5 '::chessgame)
    GROUP BY opening, board_state
    ORDER BY opening, frequency DESC
    LIMIT 5;
```

```
 opening  |                          board_state                          | frequency
----------+---------------------------------------------------------------+-----------
 1. e4 c5 | rnbqkb1r/pp2pppp/3p1n2/8/3NP3/8/PPP2PPP/RNBQKB1R w KQkq - 1 5  |       522
 1. e4 c5 | rnbqkb1r/pp1p1ppp/4pn2/8/3NP3/8/PPP2PPP/RNBQKB1R w KQkq - 1 5  |        72
 1. e4 c5 | r1bqkbnr/pp1p1ppp/2n1p3/8/3NP3/8/PPP2PPP/RNBQKB1R w KQkq - 1 5 |        69
 1. e4 c5 | r1bqkb1r/pp1ppppp/2n2n2/8/3NP3/8/PPP2PPP/RNBQKB1R w KQkq - 1 5 |        45
 1. e4 c5 | rn2kbnr/pp1qpppp/3p4/2p5/4P3/5N2/PPPP1PPP/RNBQK2R w KQkq - 0 5 |        30
(5 rows)
```

Figure 7: Most Common Board States for Specific Openings

3. How popular are the Italian Game (1.e4 e5) and the Sicilian Defense (1.e4 c5) compared to each other?

```
    SELECT
    getFirstMoves(game, 2) AS opening,
    COUNT(*) AS frequency
    FROM chessgames
    WHERE getFirstMoves(game, 2)
    IN ('1.e4 e5 '::chessgame, '1.e4 c5 '::chessgame)
    GROUP BY opening;
```

```
 opening  | frequency
----------+-----------
 1. e4 e5 |       447
 1. e4 c5 |      1080
(2 rows)
```

Figure 8: Popularity of Certain Openings

4. Get game at invalid half moves (negative half moves)

```
    SELECT id, getFirstMoves(game, -1) as move_zero
    FROM chessgames LIMIT 5;
```

```
postgres=# SELECT id, getFirstMoves(game, -1) as move_zero
FROM chessgames LIMIT 5;
  id  | move_zero
------+-----------
 1861 |
 1862 |
 1863 |
 1864 |
 1865 |
(5 rows)
```

Figure 9: Invalid half moves

5. Compare opening sequences in games and favoritegames

```
    SELECT
    g.opening_sequence,
    g.count AS games_count,
    COALESCE(fg.count, 0) AS favorite_games_count
    FROM
    (SELECT getFirstMoves(game, 2) AS opening_sequence, COUNT(*) AS
     count FROM chessgames GROUP BY opening_sequence) g
    LEFT JOIN
    (SELECT getFirstMoves(game, 2) AS opening_sequence, COUNT(*) AS
     count FROM favoritegames GROUP BY opening_sequence) fg
    ON g.opening_sequence = fg.opening_sequence;
```

opening_sequence	games_count	favorite_games_count
1. d4 Nf6	7686	0
1. e4 e5	2086	0
1. d4 e6	210	1
1. e4 c6	504	0
1. Nf3 f5	42	0
1. c4 g6	434	0
1. d4 g6	56	0
1. Nf3 g6	70	0
1. d4 c6	14	0
1. e4 Nf6	70	0
1. d4 d5	2282	0

Figure 10: Comparing opening sequences

2.3 hasBoard

1. How many games contain the given board state at any time during the game?

```
    SELECT count(*)
    FROM chessgames WHERE hasboard(game,
    'rnbqkb1r/pp1ppppp/5n2/2p1P3/8/2P5/PP1P1PPP/RNBQKBNR b KQkq - 0
     3', 5);
```

```
postgres=# SELECT count(*)
FROM chessgames
WHERE hasboard(game,
'rnbqkb1r/pp1ppppp/5n2/2p1P3/8/2P5/PP1P1PPP/RNBQKBNR b KQkq - 0 3', 5);
 count
-------
    12
(1 row)
```

Figure 11: Board state during the game

2. How many games have the initial board state but with an invalid number of moves?

```
    SELECT count(*)
    FROM chessgames
    WHERE hasBoard
    (game, 'rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0
    1', -1);
```

```
postgres=# SELECT count(*)
FROM chessgames
WHERE hasBoard(game, 'rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1', -1);
 count
-------
     0
(1 row)
```

Figure 12: Number of initial board state with an invalid move

2.4 hasOpening

1. How many games started with the given sequence of moves?

```
SELECT count(*)
FROM chessgames
WHERE hasopening(game, '1.d4 Nf6 2.Nf3 g6');
```

```
postgres=# SELECT count(*)
FROM chessgames
WHERE hasopening(game, '1.d4 Nf6 2.Nf3 g6');
 count
-------
   135
(1 row)
```

Figure 13: games starting with '1.d4 Nf6 2.Nf3 g6'

2. Which games have the same 10 first half-moves as any of the games stored in Table favoriteGames?

```
SELECT count(*)
FROM chessgames g, favoritegames f
WHERE hasopening(g.game, getFirstMoves(f.game, 10));
```

```
postgres=# SELECT count(*)
FROM chessgames g, favoritegames f
WHERE hasopening(g.game, getFirstMoves(f.game, 10));
 count
-------
    24
(1 row)
```

Figure 14: Comparison between chessgames and favoritegames table

3. How many games started with the given sequence of moves (Complete match)?

```
SELECT count(*)
FROM chessgames
WHERE hasopening(game, '1.d4 Nf6 2.Nf3 g6 3.Bg5 Bg7 4.Nbd2 O-O
5.c3 d6 6.e4 c5 7.dxc5 dxc5 8.Be2 Nc6 9.O-O b6 10.Qc2 Bb7 11.
Bh4 Nh5 12.Rfd1 Qc7 13.Nc4 Bf6 14.Ne3 e6  1/2-1/2');
```

3 Indexes

3.1 B-tree Index

Before creating the B-Tree Index:

```
EXPLAIN ANALYZE SELECT * FROM chessgames
WHERE hasOpening(game, '1.e4 c5 2.Nf3 Nc6 3.d4 cxd4');
```

```
postgres=# \COPY chessgames(game) FROM '/mnt/c/Users/Konok/Desktop/ULB-H417-PGchess-main/tests/chessgames_data.csv' DELIMITER ',' CSV;
COPY 1695
postgres=# EXPLAIN ANALYZE SELECT * FROM chessgames WHERE hasOpening(game, '1.e4 c5 2.Nf3 Nc6 3.d4 cxd4');
                                                    QUERY PLAN
-------------------------------------------------------------------------------------------------------------------
 Seq Scan on chessgames  (cost=0.00..1817.49 rows=358 width=36) (actual time=1.055..5899.606 rows=322 loops=1)
   Filter: ((game)::text ~~ '1. e4 c5 2. Nf3 Nc6 3. d4 cxd4%'::text)
   Rows Removed by Filter: 23408
 Planning Time: 0.136 ms
 Execution Time: 5899.699 ms
(5 rows)
```

Figure 15: Without B-Tree Index

1. Creating a B-tree index name **chessgames_idx** on the *game* column of *chessgame* table using **chessgame_ops** operator class. The below index is intended to optimize searching and sorting based on the *game* column.

```
CREATE INDEX hasOpening_idx ON chessgames
(CAST(game as TEXT) text_pattern_ops);
```

2. Performing cleanup of the *'chessgames'* table to recover space and update its statistical data for query optimization.

```
VACUUM ANALYZE chessgames;
```

After creating B-tree index:

```
EXPLAIN ANALYZE SELECT * FROM chessgames
WHERE hasOpening(game, '1.e4 c5 2.Nf3 Nc6 3.d4 cxd4');
```

```
postgres=# EXPLAIN ANALYZE SELECT * FROM chessgames WHERE hasOpening(game, '1.e4 c5 2.Nf3 Nc6 3.d4 cxd4');
                                                                 QUERY PLAN
---------------------------------------------------------------------------------------------------------------------------------------------
 Bitmap Heap Scan on chessgames  (cost=18.80..444.81 rows=225 width=161) (actual time=0.297..83.332 rows=322 loops=1)
   Filter: ((game)::text ~~ '1. e4 c5 2. Nf3 Nc6 3. d4 cxd4%'::text)
   Heap Blocks: exact=185
   ->  Bitmap Index Scan on hasopening_idx  (cost=0.00..18.75 rows=221 width=0) (actual time=0.140..0.140 rows=322 loops=1)
         Index Cond: (((game)::text ~>=~ '1. e4 c5 2. Nf3 Nc6 3. d4 cxd4'::text) AND ((game)::text ~<~ '1. e4 c5 2. Nf3 Nc6 3. d4 cxd5'::text))
 Planning Time: 1.082 ms
 Execution Time: 83.419 ms
(7 rows)
```

Figure 16: With B-Tree Index

3.2 Gin Index

Before creating the GIN Index:

```
1    EXPLAIN ANALYZE SELECT COUNT(*) FROM chessgames
2    WHERE hasBoard(game, 'r1bqkbnr/pppp1ppp/2n5/4p3/4P3/5N2/
     PPPP1PPP/RNBQKB1R w KQkq - 2 3', 4);
```

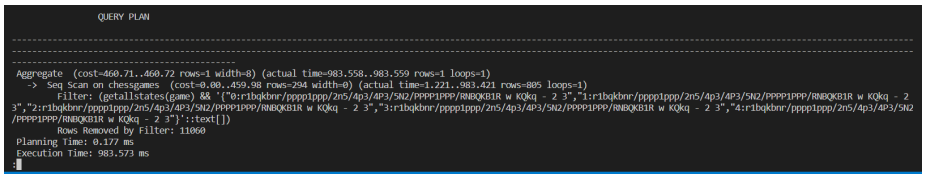

Figure 17: Without GIN Index

Creating a GIN index named *'fen_idx'* on the 'chessgames' table to improve query performance, based on the output of the *'getAllStates'* function applied to the *'game'* column.

```
1    CREATE INDEX fen_idx ON chessgames
2    USING GIN(getAllStates(game));
```

After creating GIN Index:

```
1    EXPLAIN ANALYZE SELECT COUNT(*) FROM chessgames
2    WHERE hasBoard(game, 'r1bqkbnr/pppp1ppp/2n5/4p3/4P3/5N2/
     PPPP1PPP/RNBQKB1R w KQkq - 2 3', 4);
```

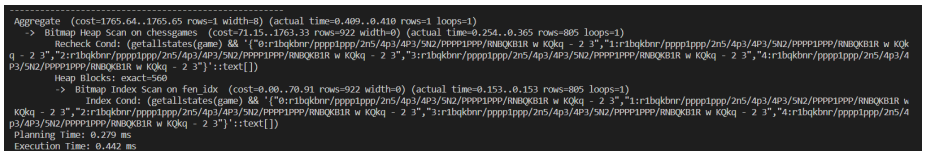

Figure 18: With GIN Index