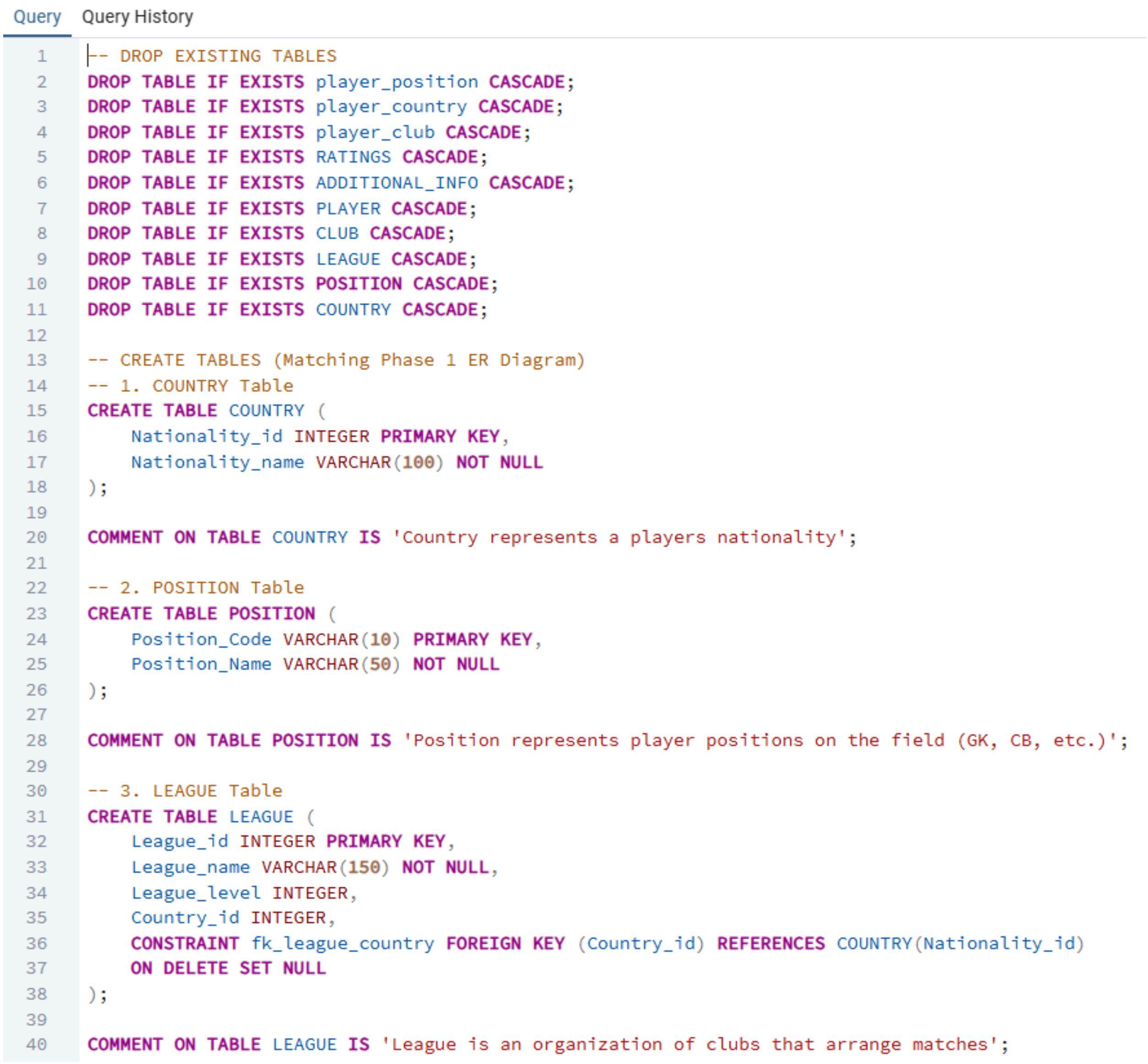
CSE412-Phase 2

Group 12 - Yichen Li, Bhavya Patel, Naimish Jayesh Maniya, Juhil Ashok Sojitra

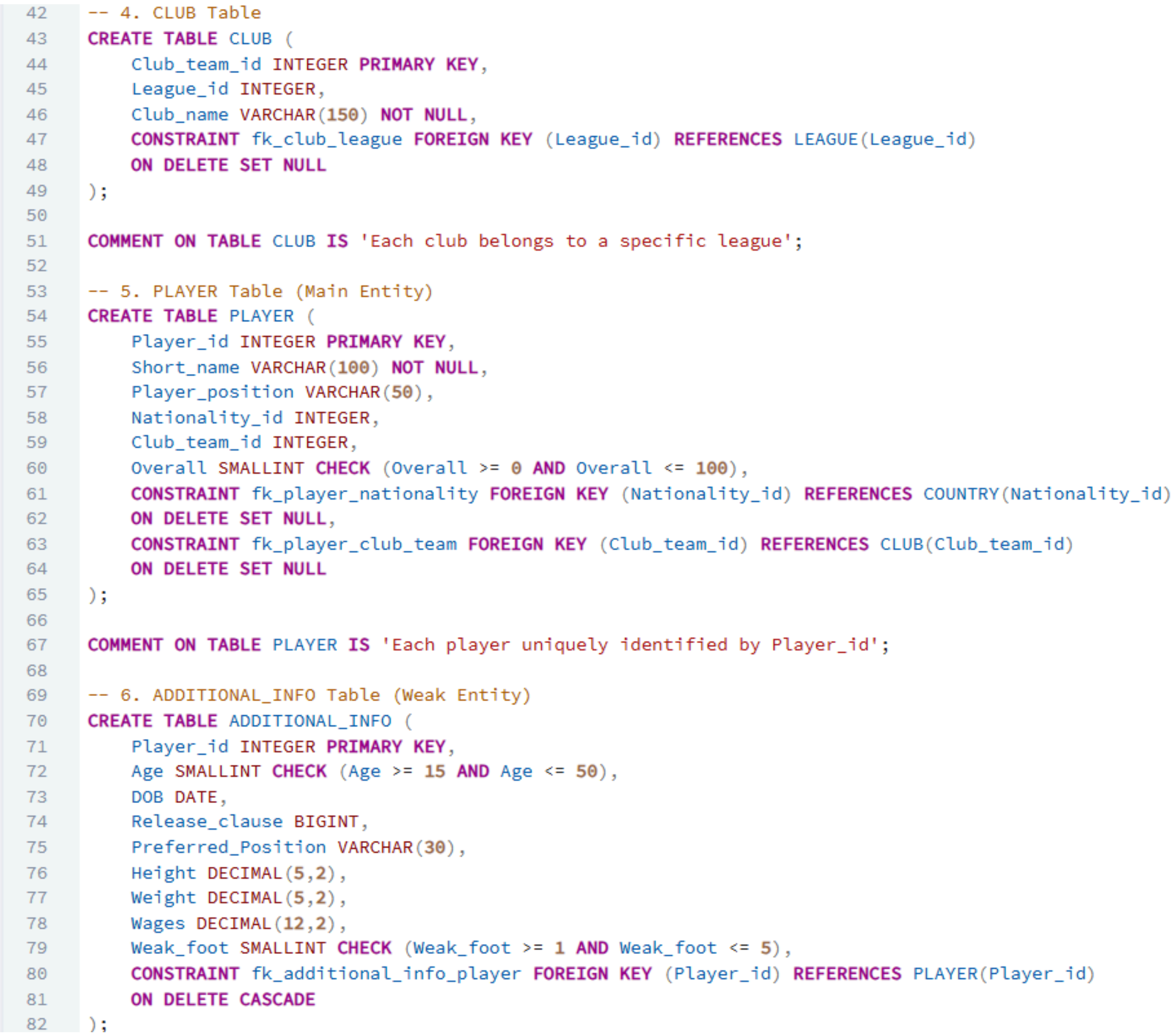
Dataset Link:
https://www.kaggle.com/datasets/rovnez/fc-26-fifa-26-player-data/data

1. ER-to-Relational Model Transformation

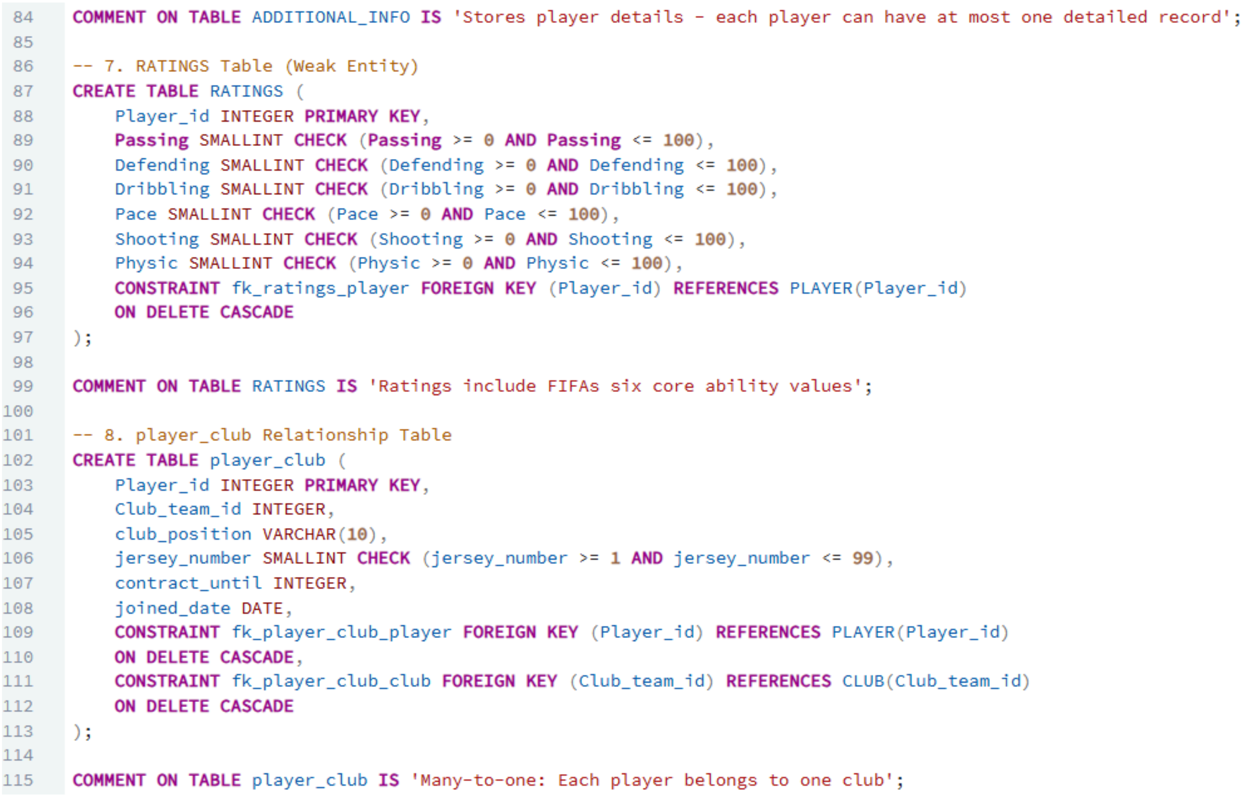
- Screenshots for DDL scripts:



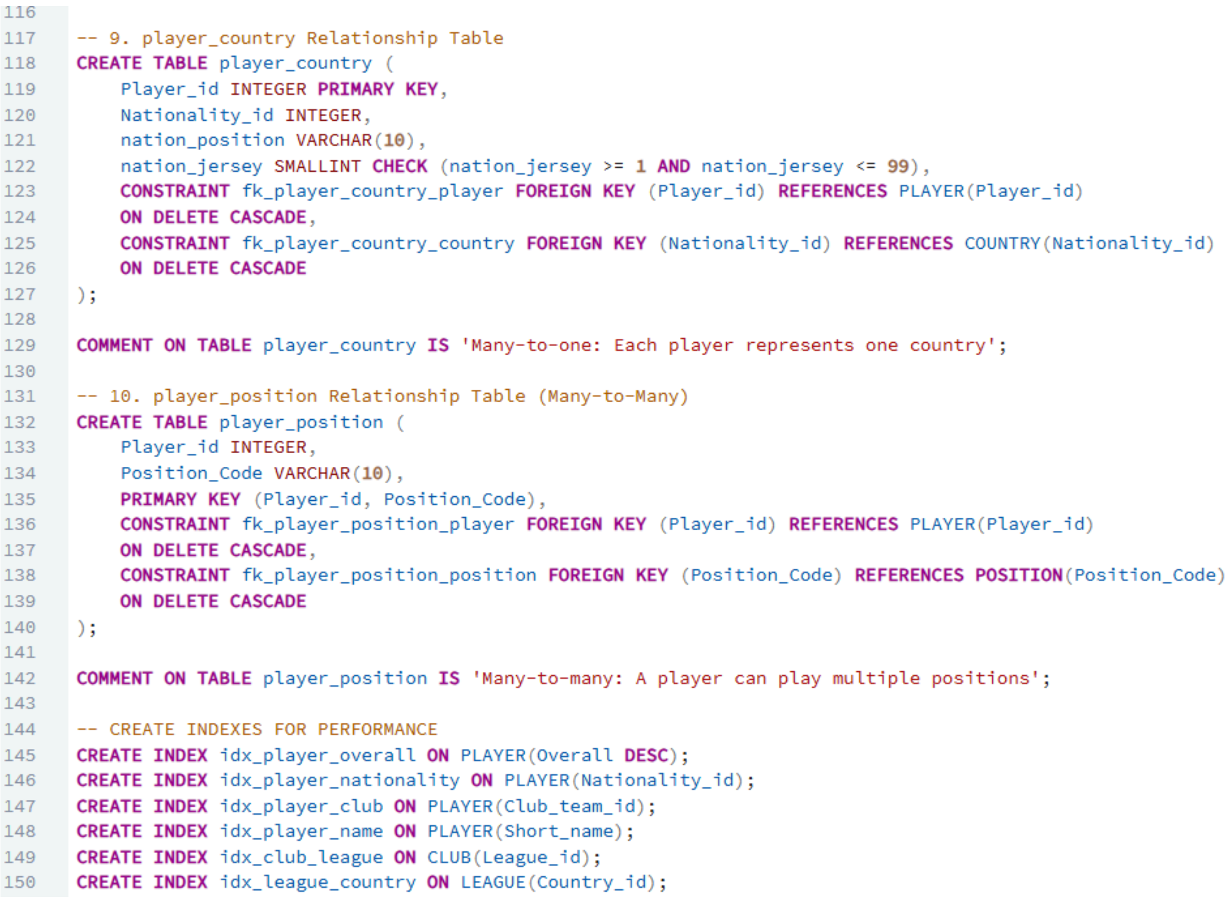
```
-- DROP EXISTING TABLES
DROP TABLE IF EXISTS player_position CASCADE;
DROP TABLE IF EXISTS player_country CASCADE;
DROP TABLE IF EXISTS player_club CASCADE;
DROP TABLE IF EXISTS RATINGS CASCADE;
DROP TABLE IF EXISTS ADDITIONAL_INFO CASCADE;
DROP TABLE IF EXISTS PLAYER CASCADE;
DROP TABLE IF EXISTS CLUB CASCADE;
DROP TABLE IF EXISTS LEAGUE CASCADE;
DROP TABLE IF EXISTS POSITION CASCADE;
DROP TABLE IF EXISTS COUNTRY CASCADE;

-- CREATE TABLES (Matching Phase 1 ER Diagram)
-- 1. COUNTRY Table
CREATE TABLE COUNTRY (
    Nationality_id INTEGER PRIMARY KEY,
    Nationality_name VARCHAR(100) NOT NULL
);

COMMENT ON TABLE COUNTRY IS 'Country represents a players nationality';

-- 2. POSITION Table
CREATE TABLE POSITION (
    Position_Code VARCHAR(10) PRIMARY KEY,
    Position_Name VARCHAR(50) NOT NULL
);

COMMENT ON TABLE POSITION IS 'Position represents player positions on the field (GK, CB, etc.)';

-- 3. LEAGUE Table
CREATE TABLE LEAGUE (
    League_id INTEGER PRIMARY KEY,
    League_name VARCHAR(150) NOT NULL,
    League_level INTEGER,
    Country_id INTEGER,
    CONSTRAINT fk_league_country FOREIGN KEY (Country_id) REFERENCES COUNTRY(Nationality_id)
    ON DELETE SET NULL
);

COMMENT ON TABLE LEAGUE IS 'League is an organization of clubs that arrange matches';
```



```sql
-- 4. CLUB Table
CREATE TABLE CLUB (
    Club_team_id INTEGER PRIMARY KEY,
    League_id INTEGER,
    Club_name VARCHAR(150) NOT NULL,
    CONSTRAINT fk_club_league FOREIGN KEY (League_id) REFERENCES LEAGUE(League_id)
    ON DELETE SET NULL
);

COMMENT ON TABLE CLUB IS 'Each club belongs to a specific league';

-- 5. PLAYER Table (Main Entity)
CREATE TABLE PLAYER (
    Player_id INTEGER PRIMARY KEY,
    Short_name VARCHAR(100) NOT NULL,
    Player_position VARCHAR(50),
    Nationality_id INTEGER,
    Club_team_id INTEGER,
    Overall SMALLINT CHECK (Overall >= 0 AND Overall <= 100),
    CONSTRAINT fk_player_nationality FOREIGN KEY (Nationality_id) REFERENCES COUNTRY(Nationality_id)
    ON DELETE SET NULL,
    CONSTRAINT fk_player_club_team FOREIGN KEY (Club_team_id) REFERENCES CLUB(Club_team_id)
    ON DELETE SET NULL
);

COMMENT ON TABLE PLAYER IS 'Each player uniquely identified by Player_id';

-- 6. ADDITIONAL_INFO Table (Weak Entity)
CREATE TABLE ADDITIONAL_INFO (
    Player_id INTEGER PRIMARY KEY,
    Age SMALLINT CHECK (Age >= 15 AND Age <= 50),
    DOB DATE,
    Release_clause BIGINT,
    Preferred_Position VARCHAR(30),
    Height DECIMAL(5,2),
    Weight DECIMAL(5,2),
    Wages DECIMAL(12,2),
    Weak_foot SMALLINT CHECK (Weak_foot >= 1 AND Weak_foot <= 5),
    CONSTRAINT fk_additional_info_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
    ON DELETE CASCADE
);
```

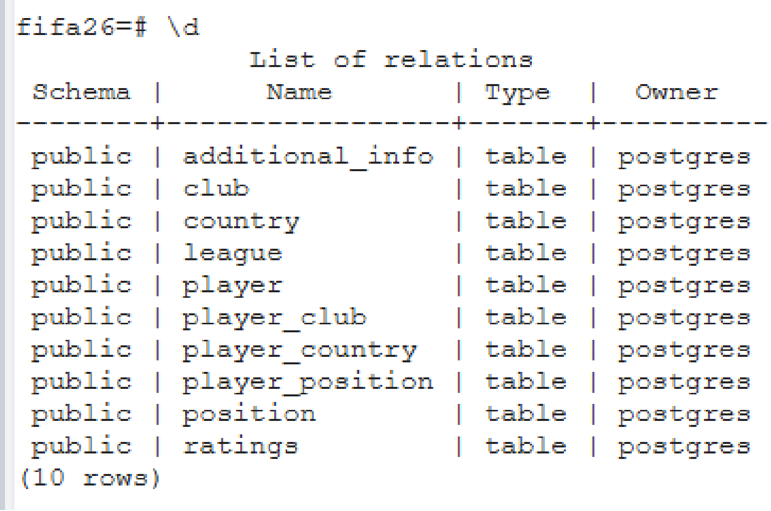


```sql
COMMENT ON TABLE ADDITIONAL_INFO IS 'Stores player details - each player can have at most one detailed record';

-- 7. RATINGS Table (Weak Entity)
CREATE TABLE RATINGS (
    Player_id INTEGER PRIMARY KEY,
    Passing SMALLINT CHECK (Passing >= 0 AND Passing <= 100),
    Defending SMALLINT CHECK (Defending >= 0 AND Defending <= 100),
    Dribbling SMALLINT CHECK (Dribbling >= 0 AND Dribbling <= 100),
    Pace SMALLINT CHECK (Pace >= 0 AND Pace <= 100),
    Shooting SMALLINT CHECK (Shooting >= 0 AND Shooting <= 100),
    Physic SMALLINT CHECK (Physic >= 0 AND Physic <= 100),
    CONSTRAINT fk_ratings_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
    ON DELETE CASCADE
);

COMMENT ON TABLE RATINGS IS 'Ratings include FIFAs six core ability values';

-- 8. player_club Relationship Table
CREATE TABLE player_club (
    Player_id INTEGER PRIMARY KEY,
    Club_team_id INTEGER,
    club_position VARCHAR(10),
    jersey_number SMALLINT CHECK (jersey_number >= 1 AND jersey_number <= 99),
    contract_until INTEGER,
    joined_date DATE,
    CONSTRAINT fk_player_club_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
    ON DELETE CASCADE,
    CONSTRAINT fk_player_club_club FOREIGN KEY (Club_team_id) REFERENCES CLUB(Club_team_id)
    ON DELETE CASCADE
);

COMMENT ON TABLE player_club IS 'Many-to-one: Each player belongs to one club';
```



```sql

-- 9. player_country Relationship Table
CREATE TABLE player_country (
    Player_id INTEGER PRIMARY KEY,
    Nationality_id INTEGER,
    nation_position VARCHAR(10),
    nation_jersey SMALLINT CHECK (nation_jersey >= 1 AND nation_jersey <= 99),
    CONSTRAINT fk_player_country_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
    ON DELETE CASCADE,
    CONSTRAINT fk_player_country_country FOREIGN KEY (Nationality_id) REFERENCES COUNTRY(Nationality_id)
    ON DELETE CASCADE
);

COMMENT ON TABLE player_country IS 'Many-to-one: Each player represents one country';

-- 10. player_position Relationship Table (Many-to-Many)
CREATE TABLE player_position (
    Player_id INTEGER,
    Position_Code VARCHAR(10),
    PRIMARY KEY (Player_id, Position_Code),
    CONSTRAINT fk_player_position_player FOREIGN KEY (Player_id) REFERENCES PLAYER(Player_id)
    ON DELETE CASCADE,
    CONSTRAINT fk_player_position_position FOREIGN KEY (Position_Code) REFERENCES POSITION(Position_Code)
    ON DELETE CASCADE
);

COMMENT ON TABLE player_position IS 'Many-to-many: A player can play multiple positions';

-- CREATE INDEXES FOR PERFORMANCE
CREATE INDEX idx_player_overall ON PLAYER(Overall DESC);
CREATE INDEX idx_player_nationality ON PLAYER(Nationality_id);
CREATE INDEX idx_player_club ON PLAYER(Club_team_id);
CREATE INDEX idx_player_name ON PLAYER(Short_name);
CREATE INDEX idx_club_league ON CLUB(League_id);
CREATE INDEX idx_league_country ON LEAGUE(Country_id);
```

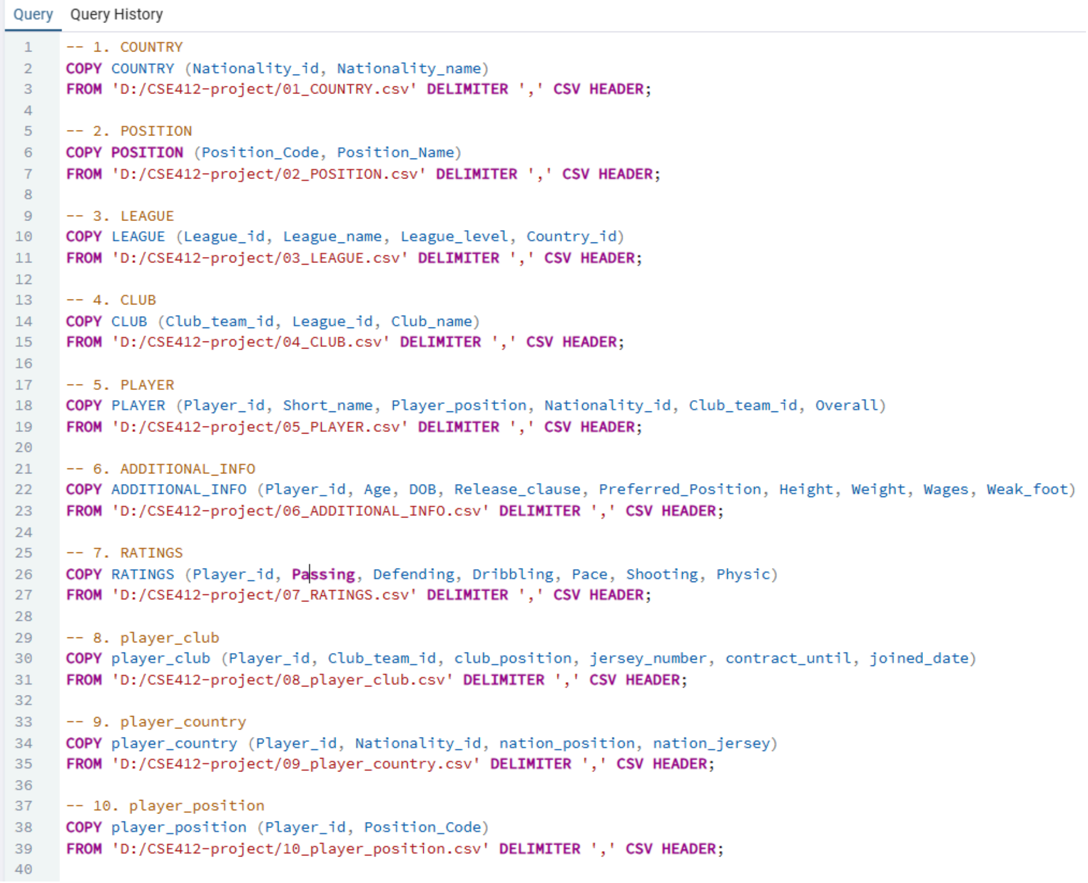
- Screenshots for schema reviews



```
fifa26=# \d
              List of relations
 Schema |       Name        | Type  |  Owner
--------+-------------------+-------+----------
 public | additional_info   | table | postgres
 public | club              | table | postgres
 public | country           | table | postgres
 public | league            | table | postgres
 public | player            | table | postgres
 public | player_club       | table | postgres
 public | player_country    | table | postgres
 public | player_position   | table | postgres
 public | position          | table | postgres
 public | ratings           | table | postgres
(10 rows)
```

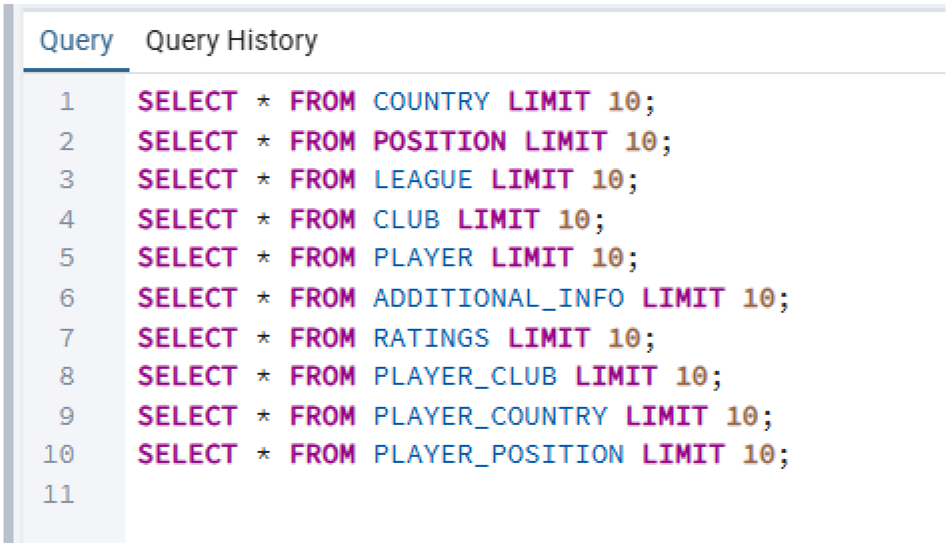
2. Data Population

- Screenshot for Copy csv to import data:



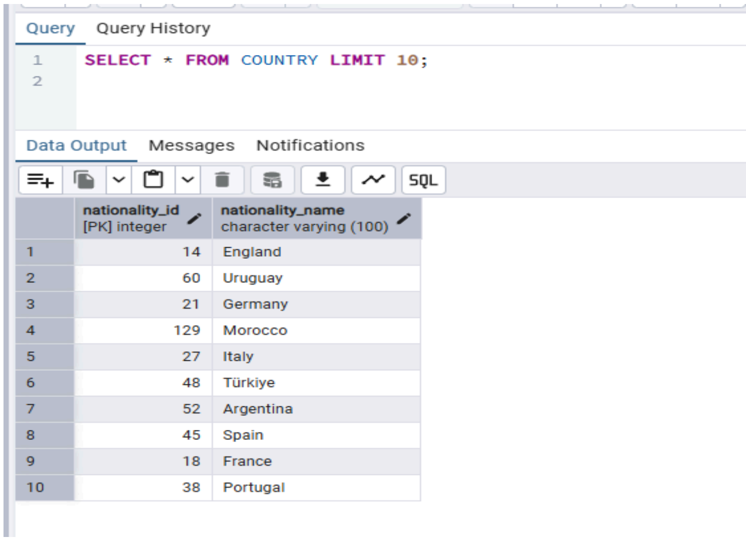
```sql
-- 1. COUNTRY
COPY COUNTRY (Nationality_id, Nationality_name)
FROM 'D:/CSE412-project/01_COUNTRY.csv' DELIMITER ',' CSV HEADER;

-- 2. POSITION
COPY POSITION (Position_Code, Position_Name)
FROM 'D:/CSE412-project/02_POSITION.csv' DELIMITER ',' CSV HEADER;

-- 3. LEAGUE
COPY LEAGUE (League_id, League_name, League_level, Country_id)
FROM 'D:/CSE412-project/03_LEAGUE.csv' DELIMITER ',' CSV HEADER;

-- 4. CLUB
COPY CLUB (Club_team_id, League_id, Club_name)
FROM 'D:/CSE412-project/04_CLUB.csv' DELIMITER ',' CSV HEADER;

-- 5. PLAYER
COPY PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
FROM 'D:/CSE412-project/05_PLAYER.csv' DELIMITER ',' CSV HEADER;

-- 6. ADDITIONAL_INFO
COPY ADDITIONAL_INFO (Player_id, Age, DOB, Release_clause, Preferred_Position, Height, Weight, Wages, Weak_foot)
FROM 'D:/CSE412-project/06_ADDITIONAL_INFO.csv' DELIMITER ',' CSV HEADER;

-- 7. RATINGS
COPY RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting, Physic)
FROM 'D:/CSE412-project/07_RATINGS.csv' DELIMITER ',' CSV HEADER;

-- 8. player_club
COPY player_club (Player_id, Club_team_id, club_position, jersey_number, contract_until, joined_date)
FROM 'D:/CSE412-project/08_player_club.csv' DELIMITER ',' CSV HEADER;

-- 9. player_country
COPY player_country (Player_id, Nationality_id, nation_position, nation_jersey)
FROM 'D:/CSE412-project/09_player_country.csv' DELIMITER ',' CSV HEADER;

-- 10. player_position
COPY player_position (Player_id, Position_Code)
FROM 'D:/CSE412-project/10_player_position.csv' DELIMITER ',' CSV HEADER;

```

- Screenshots for table contents:



```
SELECT * FROM COUNTRY LIMIT 10;
SELECT * FROM POSITION LIMIT 10;
SELECT * FROM LEAGUE LIMIT 10;
SELECT * FROM CLUB LIMIT 10;
SELECT * FROM PLAYER LIMIT 10;
SELECT * FROM ADDITIONAL_INFO LIMIT 10;
SELECT * FROM RATINGS LIMIT 10;
SELECT * FROM PLAYER_CLUB LIMIT 10;
SELECT * FROM PLAYER_COUNTRY LIMIT 10;
SELECT * FROM PLAYER_POSITION LIMIT 10;
```

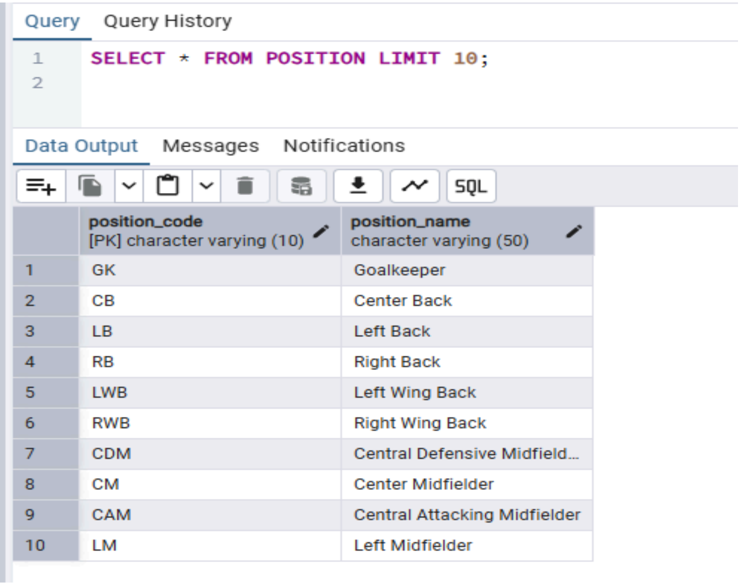
Sample 1: Country



```
SELECT * FROM COUNTRY LIMIT 10;
```

	nationality_id [PK] integer	nationality_name character varying (100)
1	14	England
2	60	Uruguay
3	21	Germany
4	129	Morocco
5	27	Italy
6	48	Türkiye
7	52	Argentina
8	45	Spain
9	18	France
10	38	Portugal

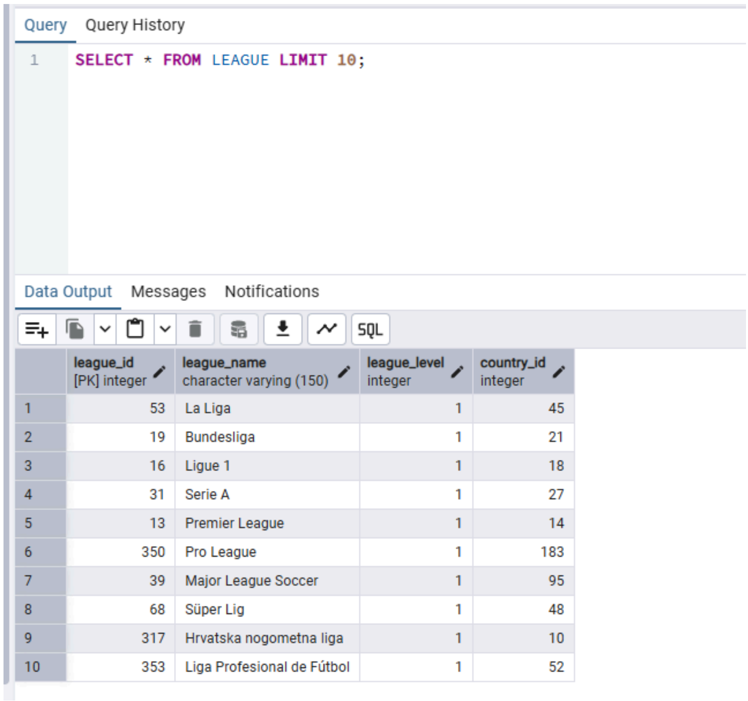
Sample 2 : Position



```sql
SELECT * FROM POSITION LIMIT 10;
```

	position_code [PK] character varying (10)	position_name character varying (50)
1	GK	Goalkeeper
2	CB	Center Back
3	LB	Left Back
4	RB	Right Back
5	LWB	Left Wing Back
6	RWB	Right Wing Back
7	CDM	Central Defensive Midfield...
8	CM	Center Midfielder
9	CAM	Central Attacking Midfielder
10	LM	Left Midfielder

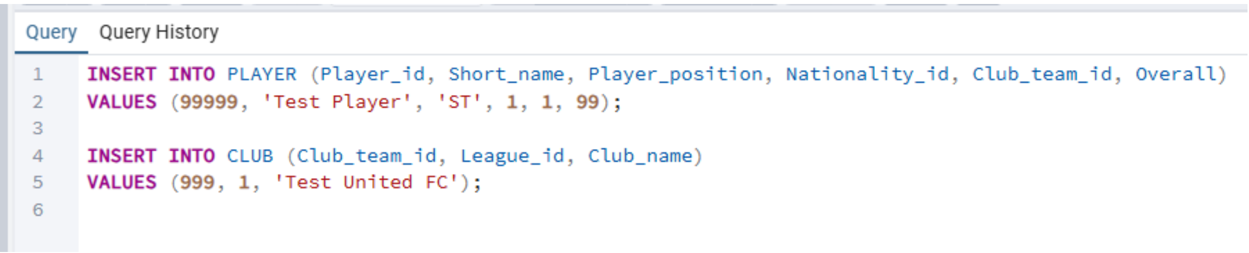
Sample 3: League



```sql
SELECT * FROM LEAGUE LIMIT 10;
```

	league_id [PK] integer	league_name character varying (150)	league_level integer	country_id integer
1	53	La Liga	1	45
2	19	Bundesliga	1	21
3	16	Ligue 1	1	18
4	31	Serie A	1	27
5	13	Premier League	1	14
6	350	Pro League	1	183
7	39	Major League Soccer	1	95
8	68	Süper Lig	1	48
9	317	Hrvatska nogometna liga	1	10
10	353	Liga Profesional de Fútbol	1	52

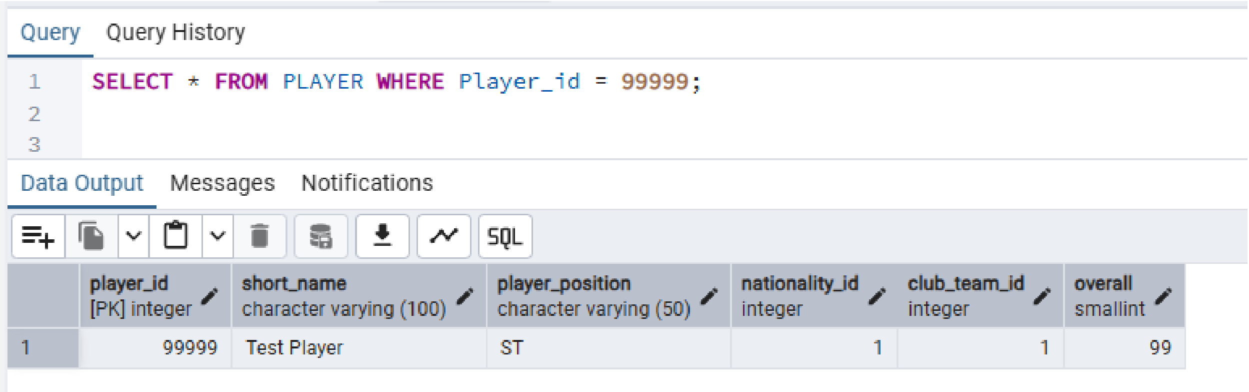
- Screenshots for sample insert commands:



```sql
INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
VALUES (99999, 'Test Player', 'ST', 1, 1, 99);

INSERT INTO CLUB (Club_team_id, League_id, Club_name)
VALUES (999, 1, 'Test United FC');

```

Sample 1 insert a player, Sample 2 insert a club.

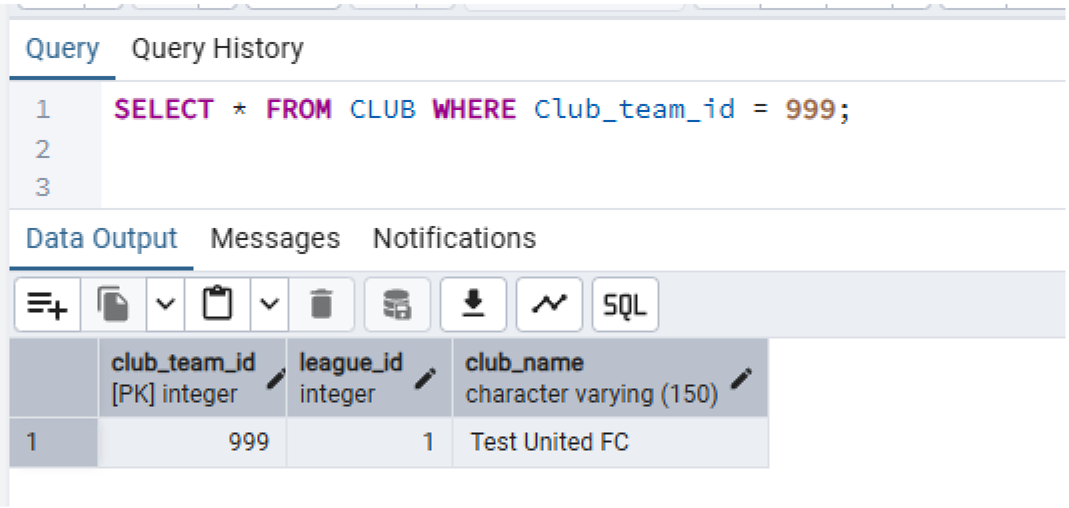
Output 1:



```sql
SELECT * FROM PLAYER WHERE Player_id = 99999;


```

	player_id [PK] integer	short_name character varying (100)	player_position character varying (50)	nationality_id integer	club_team_id integer	overall smallint
1	99999	Test Player	ST	1	1	99

Output 2:



```sql
SELECT * FROM CLUB WHERE Club_team_id = 999;


```

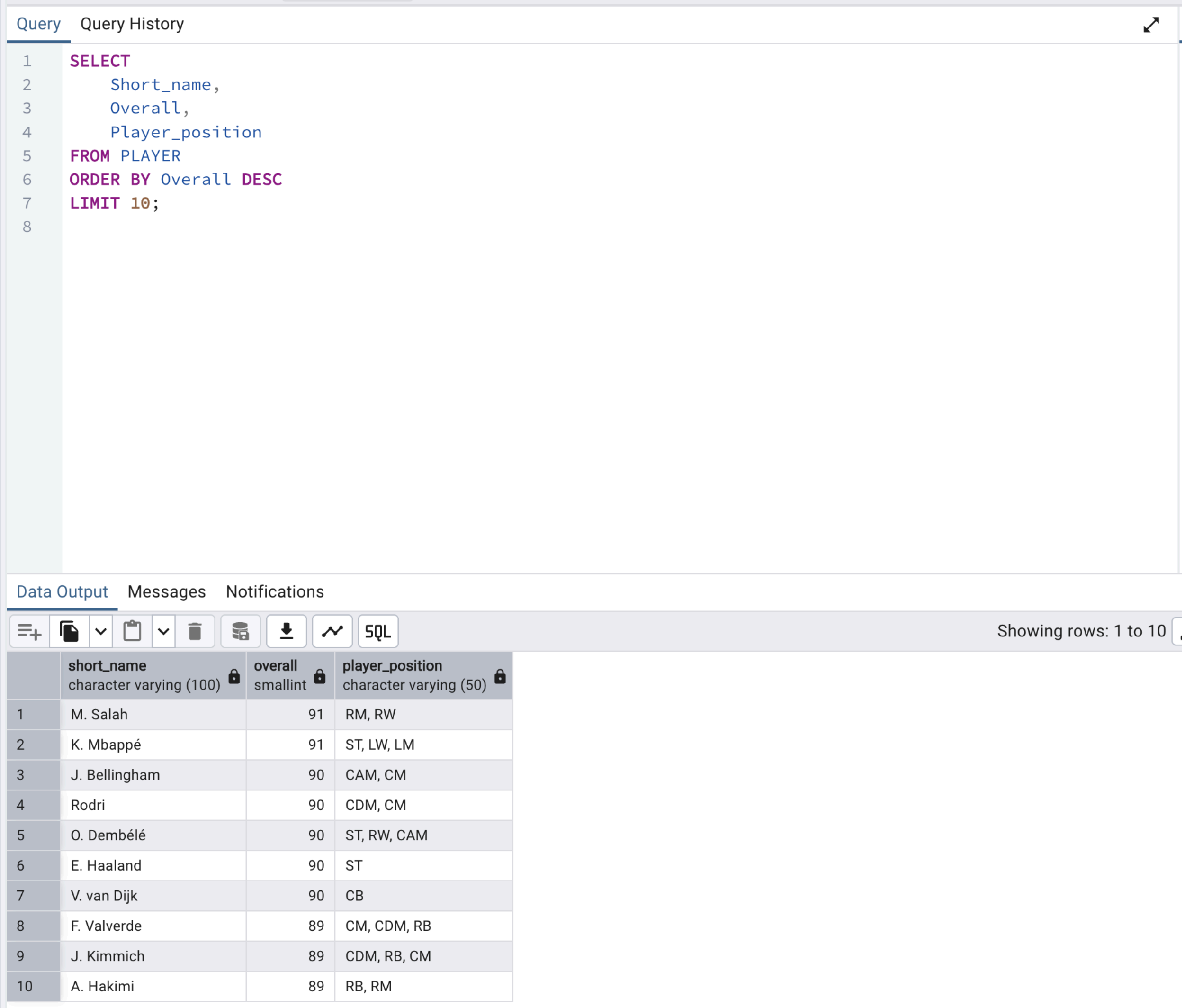
	club_team_id [PK] integer	league_id integer	club_name character varying (150)
1	999	1	Test United FC

3. SQL Queries Covering Application Use Cases

a. Query 1: SELECT - Player Search by Rating and Club

Use Case: Users search for top players from a specific club

```
SELECT
   Short_name,
   Overall,
   Player_position
FROM PLAYER
ORDER BY Overall DESC
LIMIT 10;
```



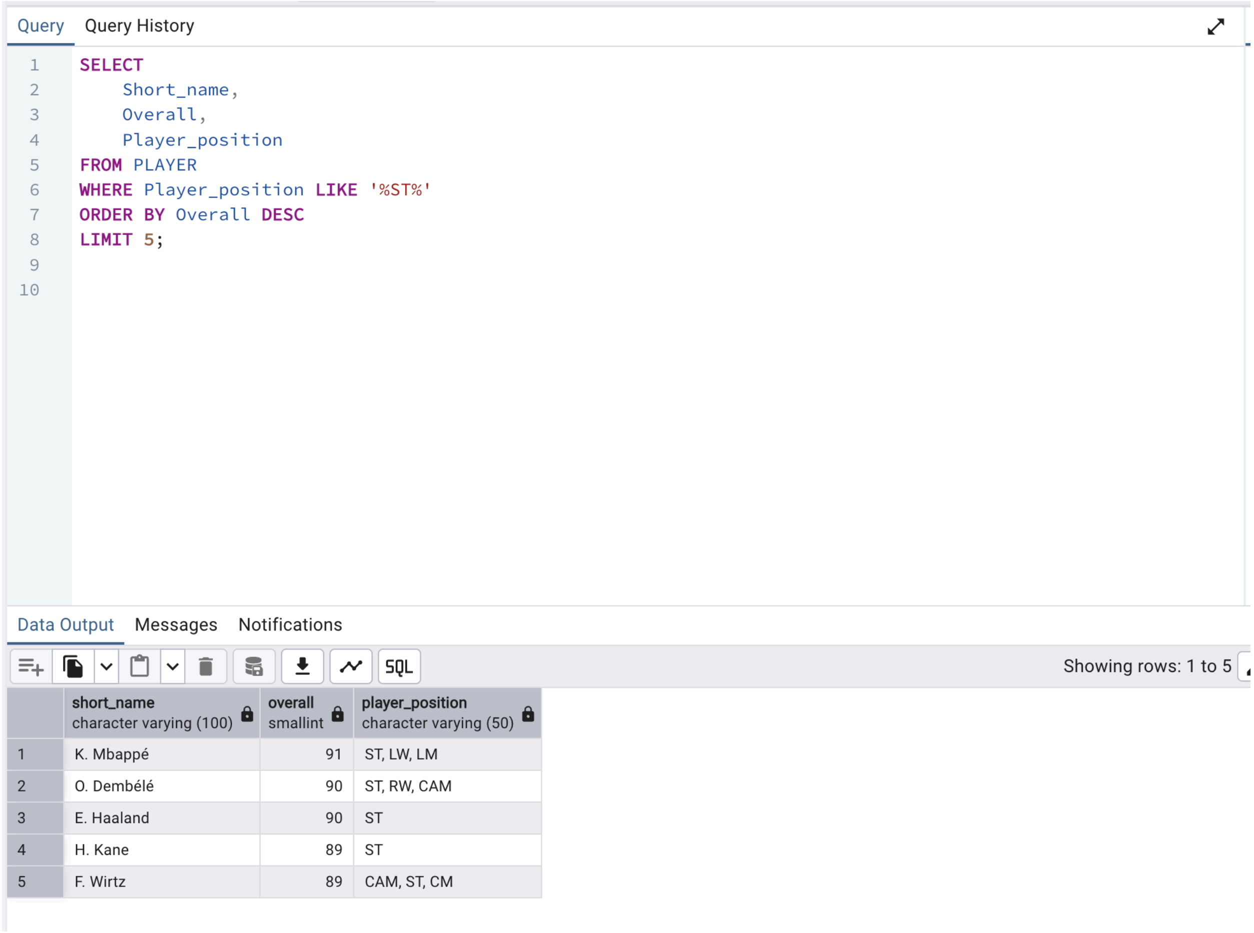
```
SELECT
    Short_name,
    Overall,
    Player_position
FROM PLAYER
ORDER BY Overall DESC
LIMIT 10;
```

	short_name character varying (100)	overall smallint	player_position character varying (50)
1	M. Salah	91	RM, RW
2	K. Mbappé	91	ST, LW, LM
3	J. Bellingham	90	CAM, CM
4	Rodri	90	CDM, CM
5	O. Dembélé	90	ST, RW, CAM
6	E. Haaland	90	ST
7	V. van Dijk	90	CB
8	F. Valverde	89	CM, CDM, RB
9	J. Kimmich	89	CDM, RB, CM
10	A. Hakimi	89	RB, RM

b. Query 2: SELECT - Player Comparison with Ratings

Use Case: Users compare multiple players side-by-side with detailed ratings

```
SELECT
    Short_name,
    Overall,
    Player_position
FROM PLAYER
WHERE Player_position LIKE '%ST%'
ORDER BY Overall DESC
LIMIT 5;
```



```
SELECT
    Short_name,
    Overall,
    Player_position
FROM PLAYER
WHERE Player_position LIKE '%ST%'
ORDER BY Overall DESC
LIMIT 5;
```

Showing rows: 1 to 5

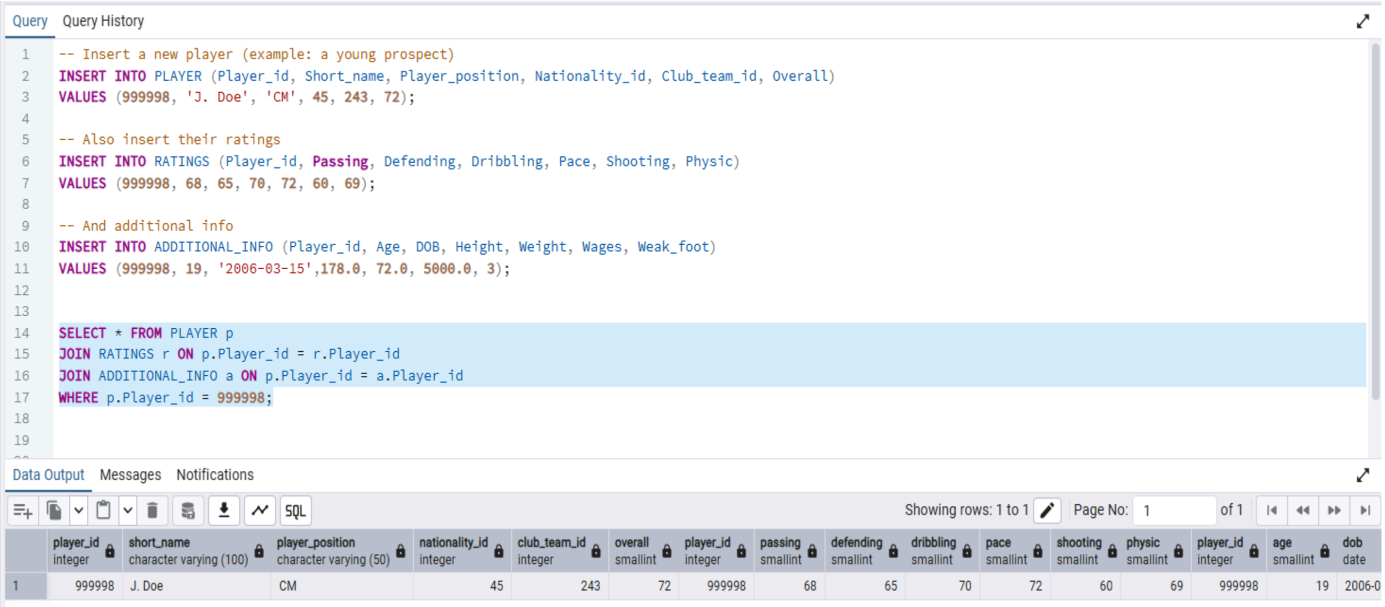
	short_name character varying (100)	overall smallint	player_position character varying (50)
1	K. Mbappé	91	ST, LW, LM
2	O. Dembélé	90	ST, RW, CAM
3	E. Haaland	90	ST
4	H. Kane	89	ST
5	F. Wirtz	89	CAM, ST, CM

c. Query 3: INSERT - Add New Player

Use Case: Add a newly signed player to the database

```
-- Insert a new player (example: a young prospect)
INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id,
Club_team_id, Overall)
VALUES (999998, 'J. Doe', 'CM', 1343, 243, 72);

-- Also insert their ratings
```

```
INSERT INTO RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting,
Physic)
VALUES (999998, 68, 65, 70, 72, 60, 69);

-- And additional info
INSERT INTO ADDITIONAL_INFO (Player_id, Age, DOB, Height, Weight, Wages,
Weak_foot)
VALUES (999998, 19, ‘2006-03-15’, 178.0, 72.0, 5000.0, 3);

-- Search for him
SELECT * FROM PLAYER p
JOIN RATINGS r ON p.Player_id = r.Player_id
JOIN ADDITIONAL_INFO a ON p.Player_id = a.Player_id
WHERE p.Player_id = 999998;
```



```
-- Insert a new player (example: a young prospect)
INSERT INTO PLAYER (Player_id, Short_name, Player_position, Nationality_id, Club_team_id, Overall)
VALUES (999998, 'J. Doe', 'CM', 45, 243, 72);

-- Also insert their ratings
INSERT INTO RATINGS (Player_id, Passing, Defending, Dribbling, Pace, Shooting, Physic)
VALUES (999998, 68, 65, 70, 72, 60, 69);

-- And additional info
INSERT INTO ADDITIONAL_INFO (Player_id, Age, DOB, Height, Weight, Wages, Weak_foot)
VALUES (999998, 19, '2006-03-15',178.0, 72.0, 5000.0, 3);


SELECT * FROM PLAYER p
JOIN RATINGS r ON p.Player_id = r.Player_id
JOIN ADDITIONAL_INFO a ON p.Player_id = a.Player_id
WHERE p.Player_id = 999998;
```

Showing rows: 1 to 1 Page No: 1 of 1

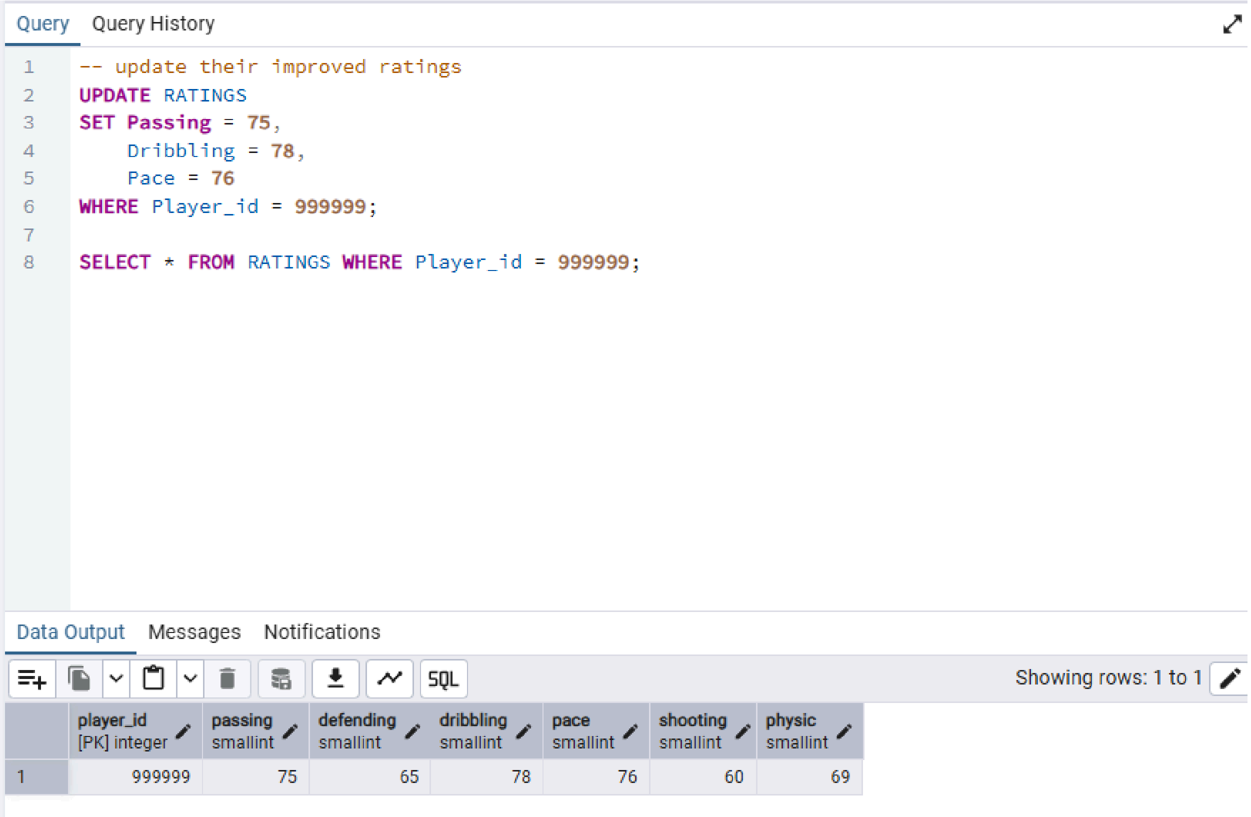
	player_id integer	short_name character varying (100)	player_position character varying (50)	nationality_id integer	club_team_id integer	overall smallint	player_id integer	passing smallint	defending smallint	dribbling smallint	pace smallint	shooting smallint	physic smallint	player_id integer	age smallint	dob date
1	999998	J. Doe	CM	45	243	72	999998	68	65	70	72	60	69	999998	19	2006-0

d. Query 4: UPDATE - Modify Player Rating

Use Case: Update player's rating after performance improvement

```
-- update their improved ratings
UPDATE RATINGS
SET Passing = 75,
   Dribbling = 78,
   Pace = 76
WHERE Player_id = 999999;
```

-- search for it
SELECT * FROM RATINGS WHERE Player_id = 999999;



```
-- update their improved ratings
UPDATE RATINGS
SET Passing = 75,
    Dribbling = 78,
    Pace = 76
WHERE Player_id = 999999;

SELECT * FROM RATINGS WHERE Player_id = 999999;
```

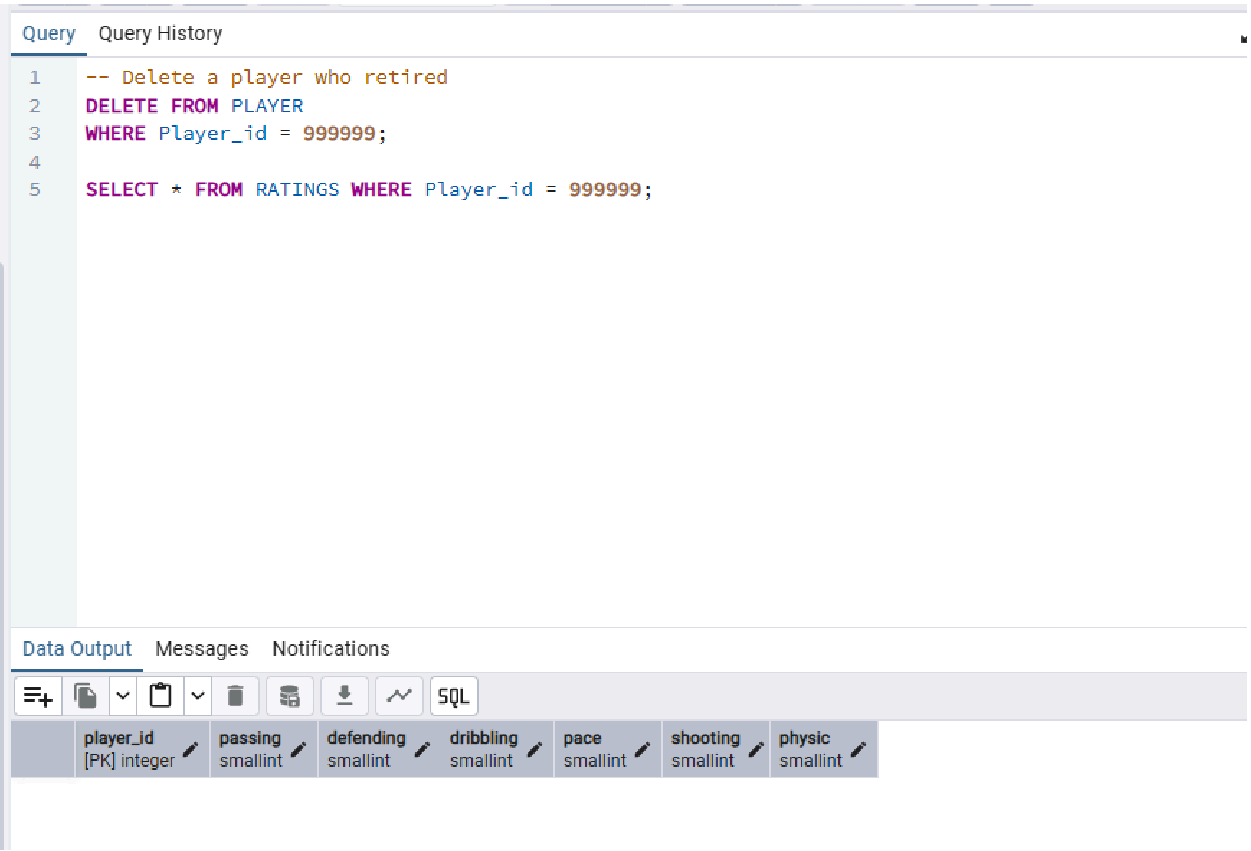
	player_id [PK] integer	passing smallint	defending smallint	dribbling smallint	pace smallint	shooting smallint	physic smallint
1	999999	75	65	78	76	60	69

e. Query 5: DELETE - Remove Player

Use Case: Remove a retired player from the database

-- delete a player who retired
DELETE FROM PLAYER
WHERE Player_id = 999999;

-- search for it
SELECT * FROM RATINGS WHERE Player_id = 999999;



```sql
-- Delete a player who retired
DELETE FROM PLAYER
WHERE Player_id = 999999;

SELECT * FROM RATINGS WHERE Player_id = 999999;
```

player_id [PK] integer	passing smallint	defending smallint	dribbling smallint	pace smallint	shooting smallint	physic smallint

4. Video

https://youtu.be/40y-hUiPa1s