Homework 2

Report

Author: Edward Student

Domain: Soccer Cup

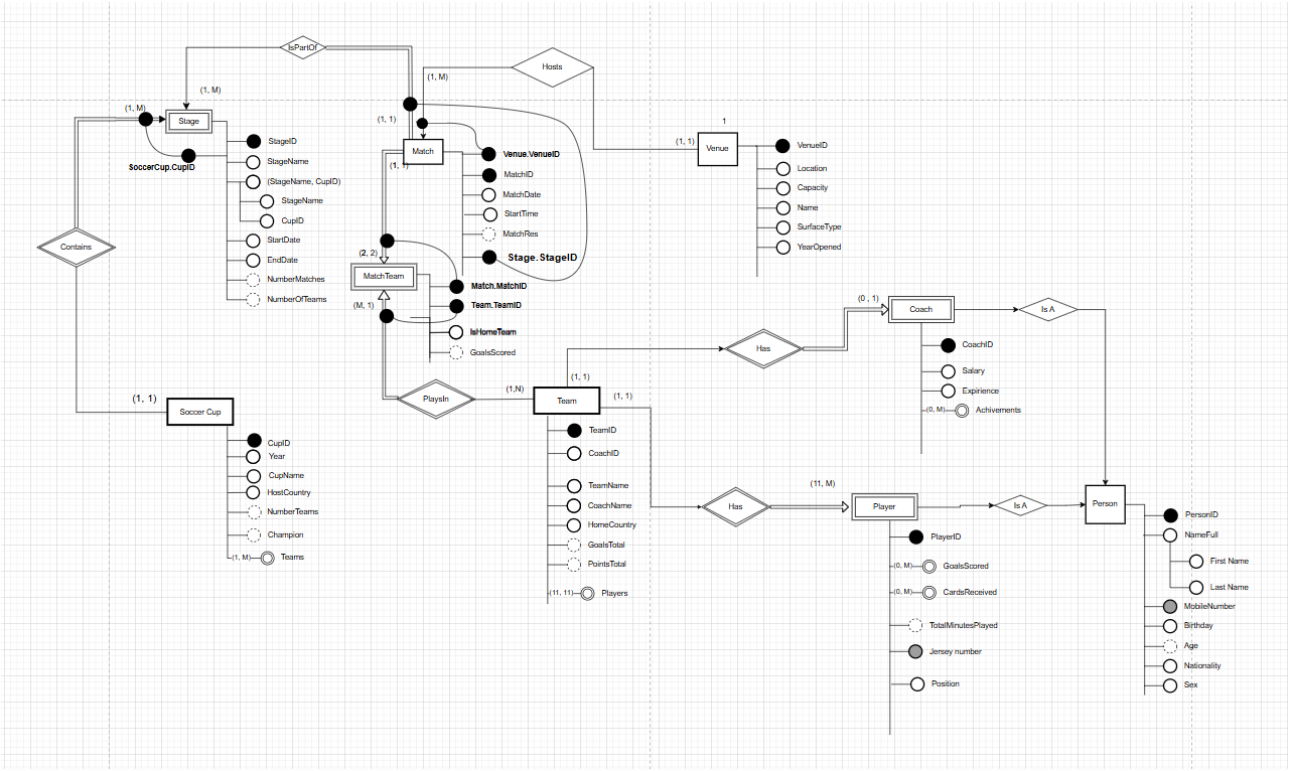
Intro step: Install MySQL on your computer

I have installed MySQl 8.4, DataGrip and MySQL Workbench 8.0 CE. The installation was understandable and easy. I made a new account and after this I create a new database and began the work

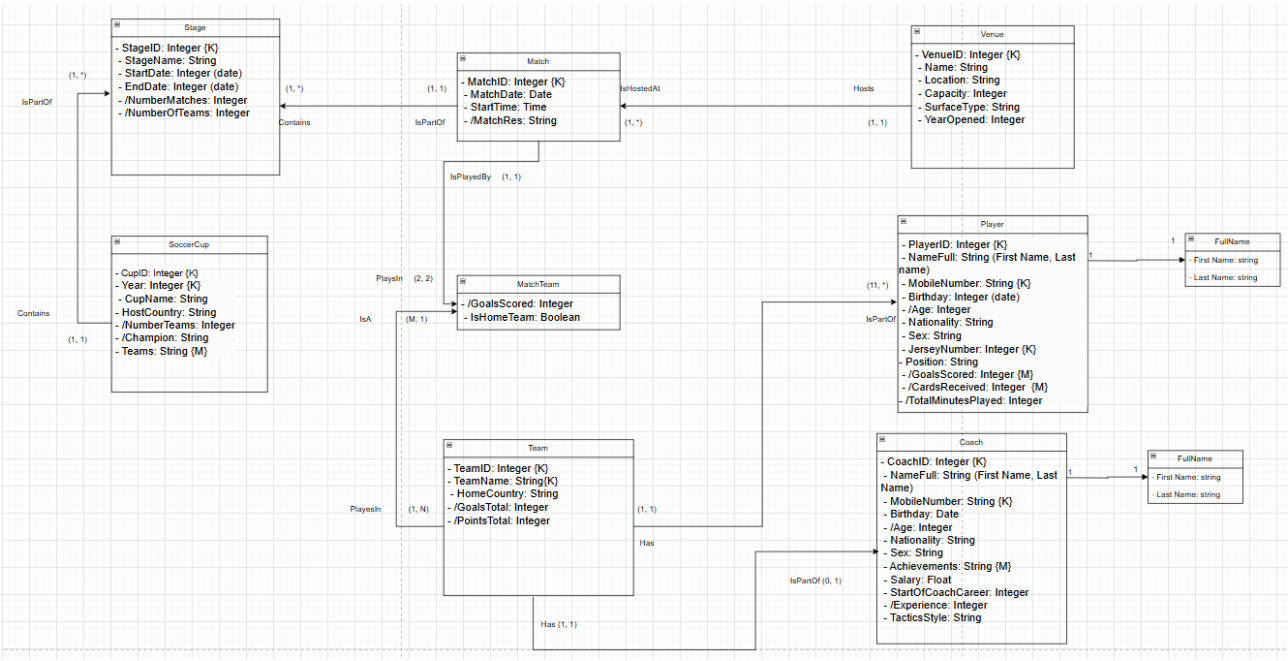
Step 1: The mapping of your conceptual model to logical model

Inputs for the step:

I have the following conceptual model:



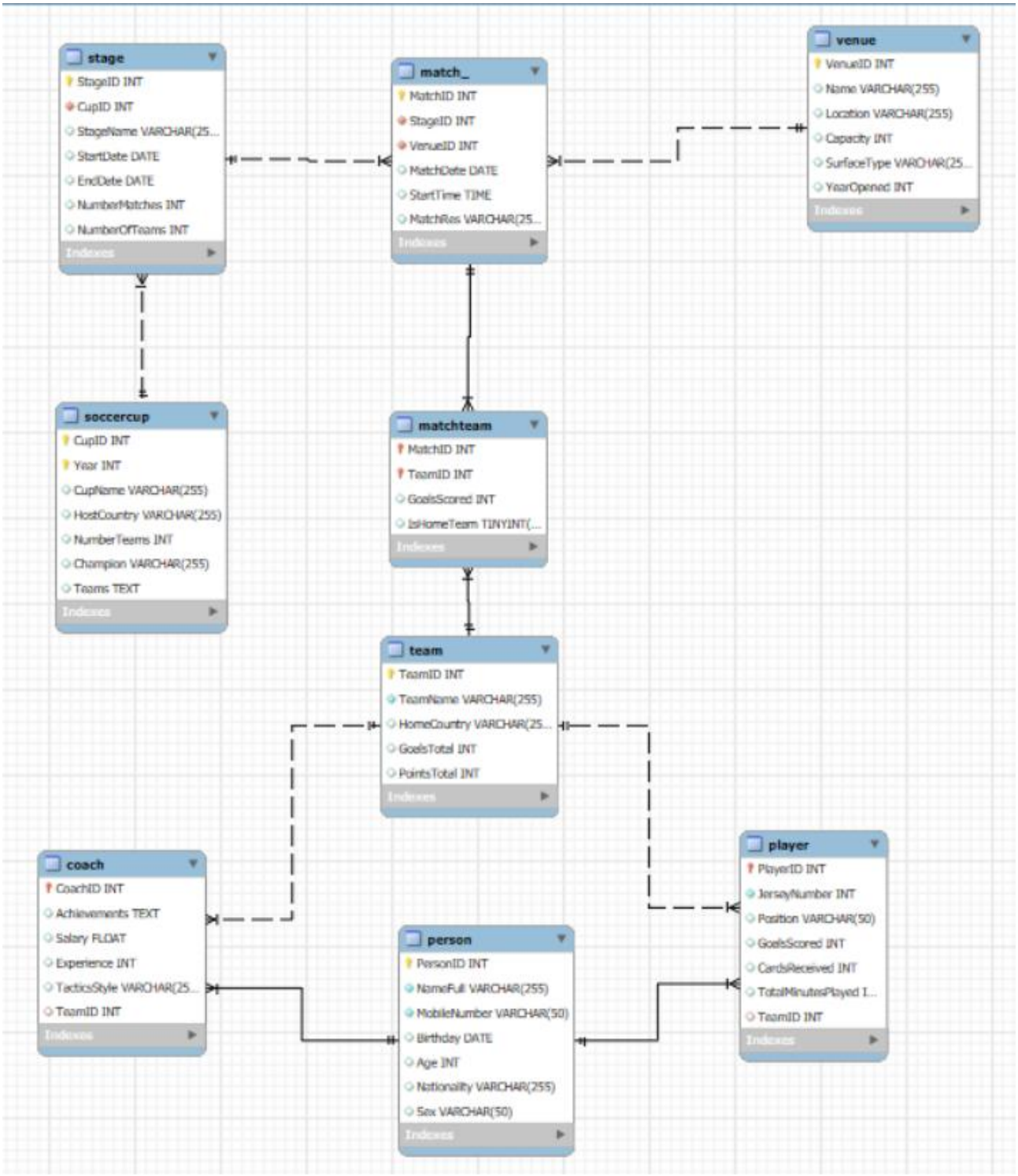
And also I have an instance from 1hw UML:



These conceptual models helped me to create a logical model in the next steps.

So I need to map the conceptual models to logical:

I created the following logical / relational model using MySQL WorkBench 8.0, based on the data from conceptual models:



Explanation of the mapping above.

Key entities and their relatioships:

1. **SoccerCup (CupID)**
    - Represents a specific soccer tournament.
    - Attributes: Year, CupName, HostCountry, Number of Teams, Champion, etc.
    - Linked to stage through CupID.

2. **Stage (StageID)**
    - Represents different stages of the tournament (e.g., group stage, knockout, final).
    - Attributes: StageName, StartDate, EndDate, Number of Matches, Number of Teams.
    - Linked to match_ through StageID.
3. **Match_ (MatchID)**

    (I changed the name of Match to Match_ because it is reserved name in DataGrip, son I cant use it)
    - Represents individual matches played in the tournament.
    - Attributes: MatchDate, StartTime, MatchRes.
    - Linked to venue through VenueID.
    - Linked to stage through StageID.
    - Linked to matchteam through MatchID.
4. **Venue (VenueID)**
    - Represents stadiums or locations where matches take place.
    - Attributes: Name, Location, Capacity, Surface Type, Year Opened.
    - Linked to match_ through VenueID.
5. **MatchTeam (MatchID, TeamID)**
    - Represents teams participating in a specific match.
    - Attributes: GoalsScored, IsHomeTeam.
    - Links match_ and team.
6. **Team (TeamID)**
    - Represents teams in the tournament.
    - Attributes: TeamName, HomeCountry, GoalsTotal, PointsTotal.
    - Linked to matchteam through TeamID.
    - Linked to coach through TeamID.
    - Linked to player through TeamID.
7. **Coach (CoachID)**
    - Represents coaches managing teams.
    - Attributes: Achievements, Salary, Experience, TacticsStyle.
    - Linked to team through TeamID.
8. **Player (PlayerID)**
    - Represents individual players in teams.
    - Attributes: JerseyNumber, Position, GoalsScored, CardsReceived, TotalMinutesPlayed.
    - Linked to team through TeamID.

- Linked to person through PersonID.

9. **Person (PersonID)**
    - Represents general personal details of a player.
    - Attributes: Name, MobileNumber, Birthday, Age, Nationality, Sex.
    - Linked to player through PersonID.

Key Mappings (Relationships)

- SoccerCup to Stage: One tournament consists of multiple stages.
- Stage to Match: Each stage includes multiple matches.
- Match to Venue: Each match is played at a specific venue.
- Match to MatchTeam: Each match involves multiple teams.
- MatchTeam to Team: Links teams to their match records.
- Team to Coach: Each team has one coach.
- Team to Player: Each team consists of multiple players.
- Player to Person: Each player is a person with personal details.

Step 2 DDL queries

Inputs :

In the step 1 I created a logical diagram which is a base to create the DDL queries for creating tables. I wrote the SQL script in the DataGrip

The provided SQL script includes DDL queries for creating tables related to a soccer tournament database. The script defines entities such as SoccerCup, Stage, Venue, Match_, Team, MatchTeam, Person, Player, and Coach. Each table is structured with appropriate attributes, primary keys (PK), and foreign keys (FK). However, the SQL code lacks a section explicitly explaining the step-by-step logic of why certain constraints or relationships were defined.

To avoid this penalty, it is recommended to provide a textual explanation preceding each DDL query to clarify its purpose and how it aligns with the logical data model.

Whether the model correscponds to logical model?

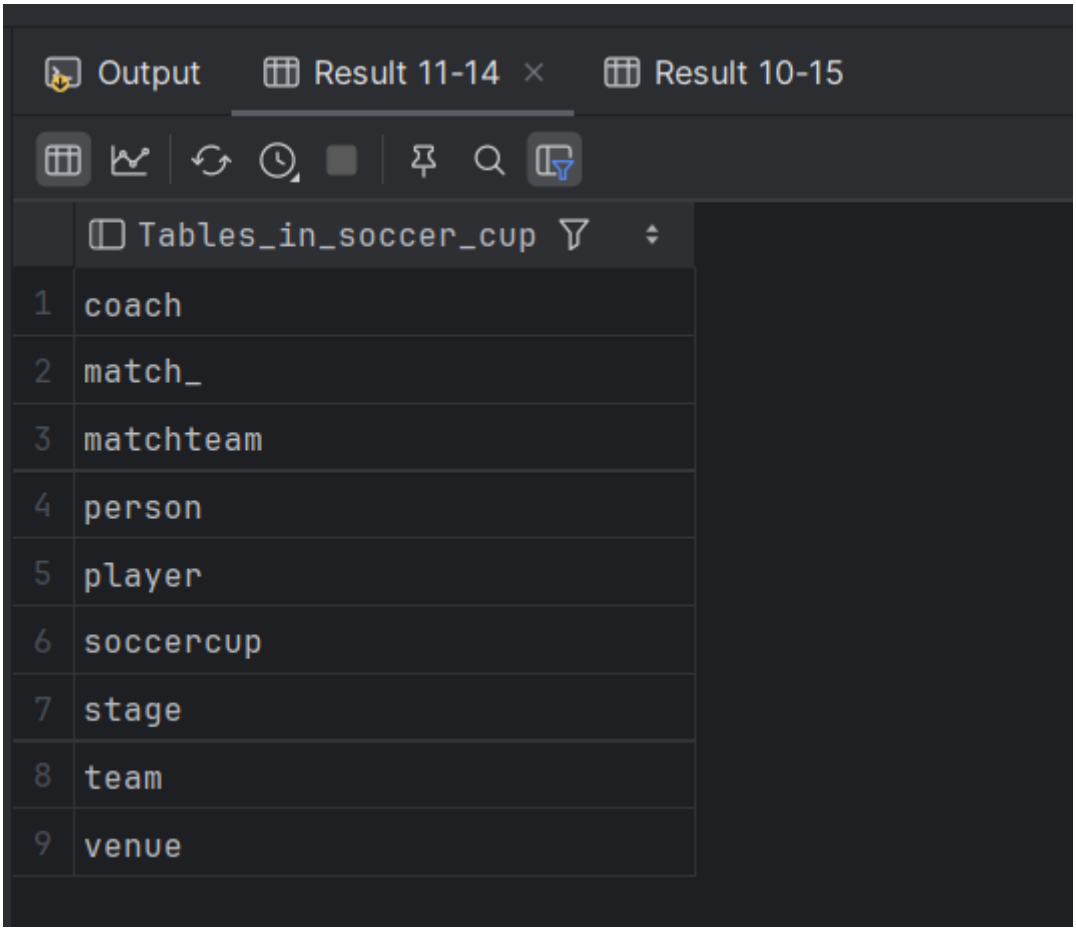
The provided SQL code closely follows the logical model presented in the logical diagram

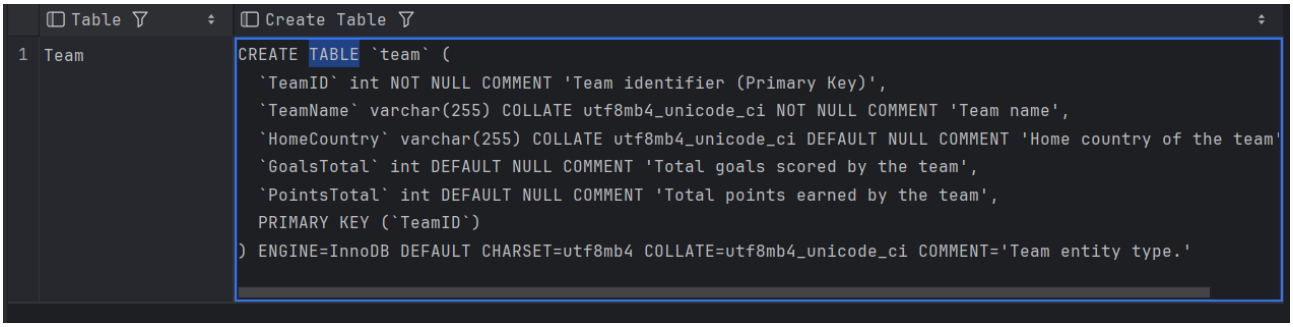
DDL Query for a Table is Not Included?

All tables presented in the ER diagram appear in the SQL script, ensuring completeness. There is no penalty for missing tables.

Important elements such a primary and foreign keys are included in the Query. All tables have the

Results per each table:





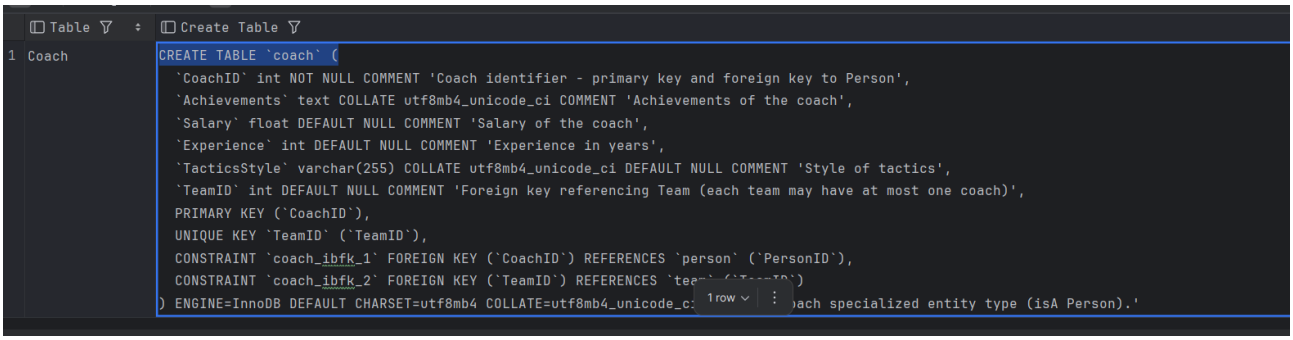


	Table	Create Table
1	Match_	CREATE TABLE `match_` (`MatchID` int NOT NULL COMMENT 'Match identifier (Primary Key)', `StageID` int NOT NULL COMMENT 'Foreign key referencing Stage', `VenueID` int NOT NULL COMMENT 'Foreign key referencing Venue', `MatchDate` date DEFAULT NULL COMMENT 'Date of the match', `StartTime` time DEFAULT NULL COMMENT 'Start time of the match', `MatchRes` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Result of the match', PRIMARY KEY (`MatchID`) COMMENT 'Primary key declaration for Match', KEY `StageID` (`StageID`), KEY `VenueID` (`VenueID`), CONSTRAINT `match__ibfk_1` FOREIGN KEY (`StageID`) REFERENCES `stage` (`StageID`), CONSTRAINT `match__ibfk_2` FOREIGN KEY (`VenueID`) REFERENCES `venue` (`VenueID`)

	Table	Create Table
1	MatchTeam	CREATE TABLE `matchteam` (`MatchID` int NOT NULL COMMENT 'Foreign key referencing Match', `TeamID` int NOT NULL COMMENT 'Foreign key referencing Team', `GoalsScored` int DEFAULT NULL COMMENT 'Goals scored in the match', `IsHomeTeam` tinyint(1) DEFAULT NULL COMMENT 'Flag indicating if the team is playing at home', PRIMARY KEY (`MatchID`,`TeamID`), KEY `TeamID` (`TeamID`), CONSTRAINT `matchteam_ibfk_1` FOREIGN KEY (`MatchID`) REFERENCES `match_` (`MatchID`), CONSTRAINT `matchteam_ibfk_2` FOREIGN KEY (`TeamID`) REFERENCES `team` (`TeamID`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='Association between Match and Team (PlaysIn relationship).'

	Table	Create Table
1	Person	CREATE TABLE `person` (`PersonID` int NOT NULL COMMENT 'Person identifier (Primary Key)', `NameFull` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT 'Full name (First Name, Last Name', `MobileNumber` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT 'Mobile number (Unique)', `Birthday` date DEFAULT NULL COMMENT 'Birthday of the person', `Age` int DEFAULT NULL COMMENT 'Age', `Nationality` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Nationality', `Sex` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Sex', PRIMARY KEY (`PersonID`), UNIQUE KEY `MobileNumber` (`MobileNumber`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='Generalized Person entity t

	Table	Create Table
1	Player	CREATE TABLE `player` (`PlayerID` int NOT NULL COMMENT 'Player identifier - primary key and foreign key to Person', `JerseyNumber` int NOT NULL COMMENT 'Jersey number (can be part of composite key if потрібно)', `Position` varchar(50) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Playing position', `GoalsScored` int DEFAULT NULL COMMENT 'Goals scored by player', `CardsReceived` int DEFAULT NULL COMMENT 'Cards received by player', `TotalMinutesPlayed` int DEFAULT NULL COMMENT 'Total minutes played', `TeamID` int DEFAULT NULL COMMENT 'Foreign key referencing Team', PRIMARY KEY (`PlayerID`), KEY `TeamID` (`TeamID`), CONSTRAINT `player_ibfk_1` FOREIGN KEY (`PlayerID`) REFERENCES `person` (`PersonID`), CONSTRAINT `player_ibfk_2` FOREIGN KEY (`TeamID`) REFERENCES `team` (`TeamID`)

Create Table: varchar(1024)

1	SoccerCup

```
CREATE TABLE `soccercup` (
  `CupID` int NOT NULL COMMENT 'Cup identifier (Part of composite PK)',
  `Year` int NOT NULL COMMENT 'Year of the cup (Part of composite PK)',
  `CupName` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Name of the cup',
  `HostCountry` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Country hosting the cup',
  `NumberTeams` int DEFAULT NULL COMMENT 'Number of teams',
  `Champion` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Champion of the cup',
  `Teams` text COLLATE utf8mb4_unicode_ci COMMENT 'List of teams participating in the cup',
  PRIMARY KEY (`CupID`,`Year`) COMMENT 'Composite primary key for Soccer Cup',
  UNIQUE KEY `uk_CupID` (`CupID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='Soccer Cup entity type.'
```

	Stage	CREATE TABLE `stage` (`StageID` int NOT NULL COMMENT 'Stage identifier (Primary Key)', `CupID` int NOT NULL COMMENT 'Foreign key to SoccerCup', `StageName` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Name of the stage', `StartDate` date DEFAULT NULL COMMENT 'Start date of the stage', `EndDate` date DEFAULT NULL COMMENT 'End date of the stage', `NumberMatches` int DEFAULT NULL COMMENT 'Number of matches', `NumberOfTeams` int DEFAULT NULL COMMENT 'Number of teams', PRIMARY KEY (`StageID`) COMMENT 'Primary key declaration for Stage', KEY `CupID` (`CupID`), CONSTRAINT `stage_ibfk_1` FOREIGN KEY (`CupID`) REFERENCES `soccercup` (`CupID`)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='Stage entity type.'

	Table	Create Table
1	Venue	CREATE TABLE `venue` (`VenueID` int NOT NULL COMMENT 'Venue identifier (Primary Key)', `Name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Name of the venue', `Location` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Location of the venue', `Capacity` int DEFAULT NULL COMMENT 'Capacity of the venue', `SurfaceType` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT 'Type of playing surface', `YearOpened` int DEFAULT NULL COMMENT 'Year the venue was opened', PRIMARY KEY (`VenueID`) COMMENT 'Primary key declaration for Venue') ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci COMMENT='Venue entity type.'

```
[2025-03-16 15:13:41] completed in 22 ms
soccer_cup> SHOW TABLES
[2025-03-16 15:13:41] 9 rows retrieved starting from 1 in 219 ms (execution: 4 ms, fetching: 215 ms)
soccer_cup> SHOW CREATE TABLE Team
[2025-03-16 15:13:41] 1 row retrieved starting from 1 in 332 ms (execution: 4 ms, fetching: 328 ms)
soccer_cup> SHOW CREATE TABLE Coach
[2025-03-16 15:13:42] 1 row retrieved starting from 1 in 372 ms (execution: 3 ms, fetching: 369 ms)
soccer_cup> SHOW CREATE TABLE Match_
[2025-03-16 15:13:42] 1 row retrieved starting from 1 in 364 ms (execution: 4 ms, fetching: 360 ms)
soccer_cup> SHOW CREATE TABLE MatchTeam
[2025-03-16 15:13:42] 1 row retrieved starting from 1 in 383 ms (execution: 3 ms, fetching: 380 ms)
soccer_cup> SHOW CREATE TABLE Person
[2025-03-16 15:13:43] 1 row retrieved starting from 1 in 368 ms (execution: 5 ms, fetching: 363 ms)
soccer_cup> SHOW CREATE TABLE Player
[2025-03-16 15:13:43] 1 row retrieved starting from 1 in 377 ms (execution: 4 ms, fetching: 373 ms)
soccer_cup> SHOW CREATE TABLE SoccerCup
[2025-03-16 15:13:43] 1 row retrieved starting from 1 in 385 ms (execution: 5 ms, fetching: 380 ms)
soccer_cup> SHOW CREATE TABLE Stage
[2025-03-16 15:13:44] 1 row retrieved starting from 1 in 392 ms (execution: 6 ms, fetching: 386 ms)
soccer_cup> SHOW CREATE TABLE Venue
[2025-03-16 15:13:44] 1 row retrieved starting from 1 in 387 ms (execution: 6 ms, fetching: 381 ms)
```

Step 3:

Populate Your Database

My input is a set of created DDL queries before and they were good for representing the logical model. Now I should to populate the database with data. For this I used cvs files, which were created for each table.

Explanations for Data Sufficiency

- **SoccerCup:** Contains data for all major soccer tournaments, ensuring the necessary CupIDs exist before inserting related tables.
- **Stage:** Links correctly to SoccerCup, ensuring no CupID is missing.

- **Venue:** Stores locations for matches, ensuring all venues are available before loading matches.
- **Match_ and MatchTeam:** Ensures all StageID and MatchID exist before referencing them.
- **Team:** Provides the teams playing in the tournament.
- **Person, Player, Coach:** Ensures people linked to teams and matches are in place.
- **Penalties:** If applicable, stores data related to match penalties.

I provided the following data files: Soc.csv, Coach.csv, Match_.csv, MatchTeam.csv, Person.csv, Player.csv, Team.csv, Venue.csv, Stage.csv. There are the files I mentioned:

Soc.csv > data

```
CupID,Year,CupName,HostCountry,NumberTeams,Champion,Teams
1,2022,World Cup,Qatar,32,Team A,Team A;Team B;Team C;Team D
2,2002,World Cup,Poland,32,Team B,Team A;Team B;Team C;Team D
3,2018,World Cup,Austria,32,Team B,Team A;Team B;Team C;Team D
```

Coach.csv > data

```
CoachID,Achievements,Salary,Experience,TacticsStyle,TeamID
1,Won League 2020,1000000,15,Attacking,1
2,Champions Cup Winner,1200000,20,Defensive,2
3,Best Coach 2022,900000,10,Possession,3
```

```
PlayerID,JerseyNumber,Position,GoalsScored,CardsReceived,TotalMinutesPlayed,TeamID
1,10,Forward,5,1,270,1
2,7,Midfielder,2,2,250,1
3,5,Defender,0,3,300,1
4,9,Forward,4,1,260,1
5,8,Midfielder,1,0,245,1
6,4,Defender,0,2,290,1
7,3,Goalkeeper,0,1,270,1
```

Match_.csv > data

```
MatchID,StageID,VenueID,MatchDate,StartTime,MatchRes
1001,1,10,2022-11-21,18:00,"2-0"
1002,1,20,2022-11-22,20:00,"1-1"
1003,2,10,2022-12-05,17:30,"0-3"
2001,1,30,2018-06-15,16:00,"1-2"
```

MatchTeam.csv > data

```csv
MatchID,TeamID,GoalsScored,IsHomeTeam
1001,1,2,1
1001,2,0,0
1002,3,1,1
1002,4,1,0
1003,2,0,1
1003,3,3,0
2001,1,1,1
2001,4,2,0
```

Person.csv > data

```csv
PersonID,NameFull,MobileNumber,Birthday,Age,Nationality,Sex
1,John Doe,+1234567890,1990-05-20,34,Country X,Male
2,Jane Smith,+0987654321,1985-07-15,38,Country Y,Female
3,David Johnson,+1122334455,1995-09-10,29,Country Z,Male
4,Chris Brown,+1212121212,1998-03-25,26,Country X,Male
5,Lisa Adams,+1313131313,1993-06-14,31,Country Y,Female
6,Tom Clark,+1414141414,2000-01-30,24,Country Z,Male
7,Emma Wilson,+1515151515,1991-09-12,33,Country X,Female
8,Michael Lee,+1616161616,1997-11-21,27,Country Y,Male
9,Sarah Kim,+1717171717,1994-07-08,30,Country Z,Female
10,James White,+1818181818,1992-05-05,32,Country X,Male
11,Olivia Martinez,+1919191919,1996-12-17,28,Country Y,Female
12,Daniel Harris,+2020202020,1999-04-22,25,Country Z,Male
13,Sophia Anderson,+2121212121,1995-08-19,29,Country X,Female
14,Ethan Scott,+2222222222,1990-02-15,34,Country Y,Male
15,Mia Thompson,+2323232323,1988-10-03,36,Country Z,Female
16,Alexander Carter,+2424242424,2001-06-30,23,Country X,Male
17,Charlotte Evans,+2525252525,1992-01-11,32,Country Y,Female
18,Benjamin Parker,+2626262626,1993-11-09,31,Country Z,Male
19,Ava Rodriguez,+2727272727,1997-02-26,27,Country X,Female
20,William Green,+2828282828,2002-12-01,22,Country Y,Male
21,Emily Lewis,+2929292929,1994-09-07,30,Country Z,Female
22,Henry Walker,+3030303030,1998-07-16,26,Country X,Male
```

Stage.csv > data

```csv
StageID,CupID,StageName,StartDate,EndDate,NumberMatches,NumberOfTeams
1,1,Group Stage,2024-06-10,2024-06-25,36,32
2,1,Round of 16,2024-06-26,2024-06-29,8,16
3,1,Quarter-finals,2024-06-30,2024-07-03,4,8
4,1,Semi-finals,2024-07-04,2024-07-06,2,4
5,1,Final,2024-07-07,2024-07-07,1,2
```

Team.csv > data

```
TeamID,TeamName,HomeCountry,GoalsTotal,PointsTotal
1,"Team A","Country A",45,20
2,"Team B","Country B",50,22
3,"Team C","Country C",37,18
4,"Team D","Country D",40,19
```

Venue.csv > data

```
VenueID,Name,Location,Capacity,SurfaceType,YearOpened
10,"Lusail Iconic Stadium","Lusail",80000,"Grass",2020
20,"Al Bayt Stadium","Al Khor",60000,"Grass",2019
30,"Kolos","Terebovlia",68000,"Grass",2007
```

I created a new sql file: import_data.sql, to write the queries into this file for convinient import of data from csv files.

There is an example of a population query, which take the data from the .csv file for Soccer Cup:

```
LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Soc.csv'
INTO TABLE SoccerCup
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

SELECT * FROM SoccerCup;
```

Other tables are filled as well.

```sql
LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Stage.csv'
INTO TABLE Stage
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
SELECT * FROM Stage;


LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Venue.csv'
INTO TABLE Venue
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
SELECT * FROM Venue;

LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Match_.csv'
INTO TABLE Match_
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
SELECT * FROM Match_;
```

```sql
LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Team.csv'
INTO TABLE Team
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
SELECT * FROM Team;

LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/MatchTeam.csv'
INTO TABLE MatchTeam
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
SELECT * FROM matchteam;

LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Person.csv'
INTO TABLE Person
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

```
70
71 ✓ LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Player.csv'
72   INTO TABLE Player
73   FIELDS TERMINATED BY ','
74   OPTIONALLY ENCLOSED BY '"'
75   LINES TERMINATED BY '\r\n'
76   IGNORE 1 LINES;
77 ✓ SELECT * FROM Player;
78
79 ✓ LOAD DATA LOCAL INFILE 'C:/Users/user/Documents/year2sem2/databases/hw2cvs/Coach.csv'
80   INTO TABLE Coach
81   FIELDS TERMINATED BY ','
82   OPTIONALLY ENCLOSED BY '"'
83   LINES TERMINATED BY '\r\n'
84   IGNORE 1 LINES;
85 ✓ SELECT * FROM Coach;
```

Here are results of getting the data from the csv. An order of the results is similar to the order of popupate queries above.

	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D

	StageID	CupID	StageName	StartDate	EndDate	NumberMatches	NumberOfTeams
1	1	1	Group Stage	2024-06-10	2024-06-25	16	32
2	2	1	Round of 16	2024-06-26	2024-06-29	8	16
3	3	1	Quarter-finals	2024-06-30	2024-07-03	4	8
4	4	1	Semi-finals	2024-07-04	2024-07-06	2	4
5	5	1	Final	2024-07-07	2024-07-07	1	2

	VenueID	Name	Location	Capacity	SurfaceType	YearOpened
1	10	Lusail Iconic Stadium	Lusail	80000	Grass	2020
2	20	Al Bayt Stadium	Al Khor	60000	Grass	2019
3	30	Kolos	Terebovlia	68000	Grass	2007

	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	10	2022-11-21	18:00:00	2-0
2	1002	1	20	2022-11-22	20:00:00	1-1
3	1003	2	10	2022-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

	TeamID	TeamName	HomeCountry	GoalsTotal	PointsTotal
1	1	Team A	Country A	45	20
2	2	Team B	Country B	50	22
3	3	Team C	Country C	37	18
4	4	Team D	Country D	40	19

	MatchID	TeamID	GoalsScored	IsHomeTeam
1	1001	1	2	1
2	1001	2	0	0
3	1002	3	1	1
4	1002	4	1	0
5	1003	2	0	1
6	1003	3	3	0
7	2001	1	1	1
8	2001	4	2	0

WHERE ORDER BY

	PersonID	NameFull	MobileNumber	Birthday	Age	Nationality	Sex
1	1	John Doe	+1234567890	1990-05-20	34	Hungarian	Male
2	2	Jane Smith	+0987654321	1985-07-15	38	Italian	Female
3	3	David Johnson	+1122334455	1995-09-10	29	Ukrainian	Male
4	4	Chris Brown	+1212121212	1998-03-25	26	Hungarian	Male
5	5	Lisa Adams	+1313131313	1993-06-14	31	Italian	Female
6	6	Tom Clark	+1414141414	2000-01-30	24	Ukrainian	Male
7	7	Emma Wilson	+1515151515	1991-09-12	33	Hungarian	Female
8	8	Michael Lee	+1616161616	1997-11-21	27	Italian	Male
9	9	Sarah Kim	+1717171717	1994-07-08	30	Ukrainian	Female
10	10	James White	+1818181818	1992-05-05	32	Hungarian	Male
11	11	Olivia Martinez	+1919191919	1996-12-17	28	Italian	Female
12	12	Daniel Harris	+2020202020	1999-04-22	25	Ukrainian	Male
13	13	Sophia Anderson	+2121212121	1995-08-19	29	Hungarian	Female
14	14	Ethan Scott	+2222222222	1990-02-15	34	Italian	Male

	PlayerID	JerseyNumber	Position	GoalsScored	CardsReceived	TotalMinutesPlayed	TeamID
1	1	10	Forward	5	1	270	1
2	2	7	Midfielder	2	2	250	1
3	3	5	Defender	0	3	300	1
4	4	9	Forward	4	1	260	1
5	5	8	Midfielder	1	0	245	1
6	6	4	Defender	0	2	290	1
7	7	3	Goalkeeper	0	1	270	1
8	8	11	Forward	3	0	255	2
9	9	6	Midfielder	2	2	275	2
10	10	2	Defender	0	1	280	2
11	11	1	Goalkeeper	0	0	270	2
12	12	12	Forward	5	1	265	2
13	13	14	Midfielder	2	2	250	2
14	14	13	Defender	0	3	300	2
15	15	16	Forward	4	1	260	3

	CoachID	Achievements	Salary	Experience	TacticsStyle	TeamID
1	1	Won League 2020	1000000	15	Attacking	1
2	2	Champions Cup Winner	1200000	20	Defensive	2
3	3	Best Coach 2022	900000	10	Possession	3

Step 4:

Select data

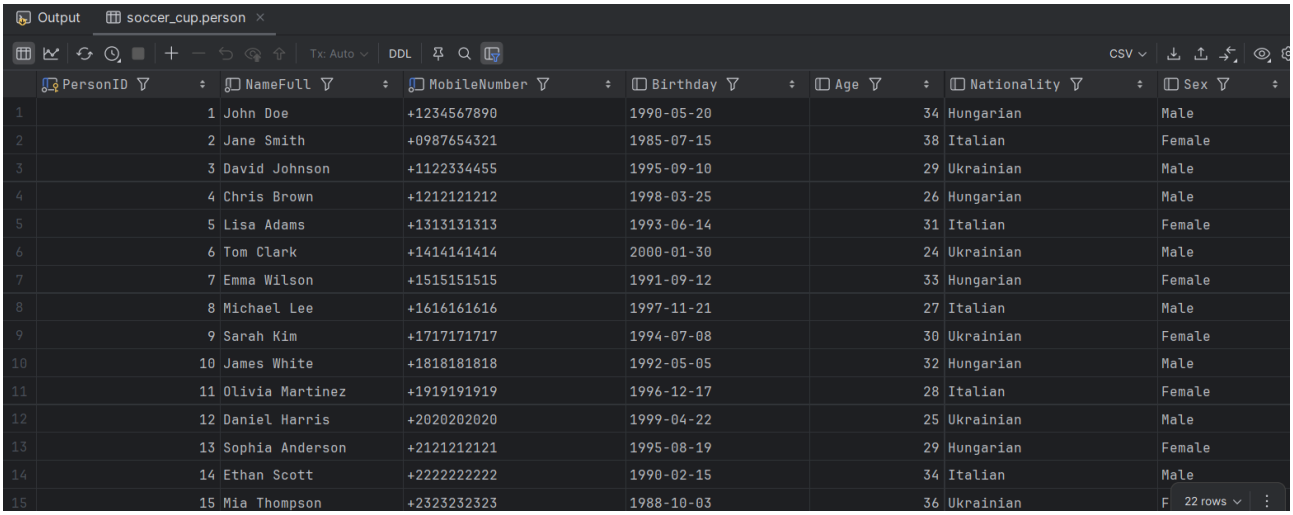
At the input, I have created tables which are filled with the data obtained from csv. There is a lot of data which I can test. You can see source data (input) below in examples

I used the following order to show you work: Source data -> query -> Result

Union query:

I want to get all persons who are Hungarian and Ukrainian female:

Source data:



Output | soccer_cup.person

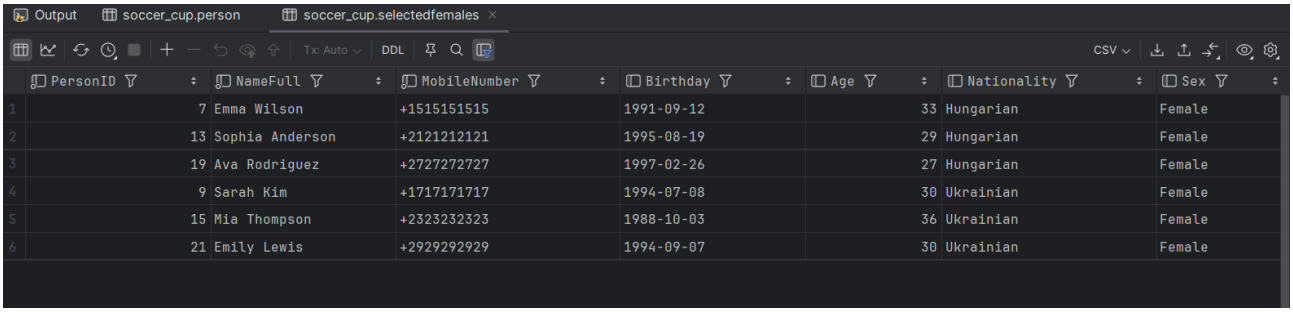
	PersonID	NameFull	MobileNumber	Birthday	Age	Nationality	Sex
1	1	John Doe	+1234567890	1990-05-20	34	Hungarian	Male
2	2	Jane Smith	+0987654321	1985-07-15	38	Italian	Female
3	3	David Johnson	+1122334455	1995-09-10	29	Ukrainian	Male
4	4	Chris Brown	+1212121212	1998-03-25	26	Hungarian	Male
5	5	Lisa Adams	+1313131313	1993-06-14	31	Italian	Female
6	6	Tom Clark	+1414141414	2000-01-30	24	Ukrainian	Male
7	7	Emma Wilson	+1515151515	1991-09-12	33	Hungarian	Female
8	8	Michael Lee	+1616161616	1997-11-21	27	Italian	Male
9	9	Sarah Kim	+1717171717	1994-07-08	30	Ukrainian	Female
10	10	James White	+1818181818	1992-05-05	32	Hungarian	Male
11	11	Olivia Martinez	+1919191919	1996-12-17	28	Italian	Female
12	12	Daniel Harris	+2020202020	1999-04-22	25	Ukrainian	Male
13	13	Sophia Anderson	+2121212121	1995-08-19	29	Hungarian	Female
14	14	Ethan Scott	+2222222222	1990-02-15	34	Italian	Male
15	15	Mia Thompson	+2323232323	1988-10-03	36	Ukrainian	F

22 rows

Union Query:

```
129 ✓  SELECT * FROM person;
130
131 ✓  DROP TABLE IF EXISTS SelectedFemales;
132
133 ✓  CREATE TABLE SelectedFemales AS
134    SELECT * FROM Person WHERE Nationality = 'Hungarian' AND Sex = 'Female'
135    UNION
136    SELECT  PersonID * FROM Person WHERE Nationality = 'Ukrainian' AND Sex = 'Female';
137
138 ✓  SELECT * FROM SelectedFemales;
```

The results:

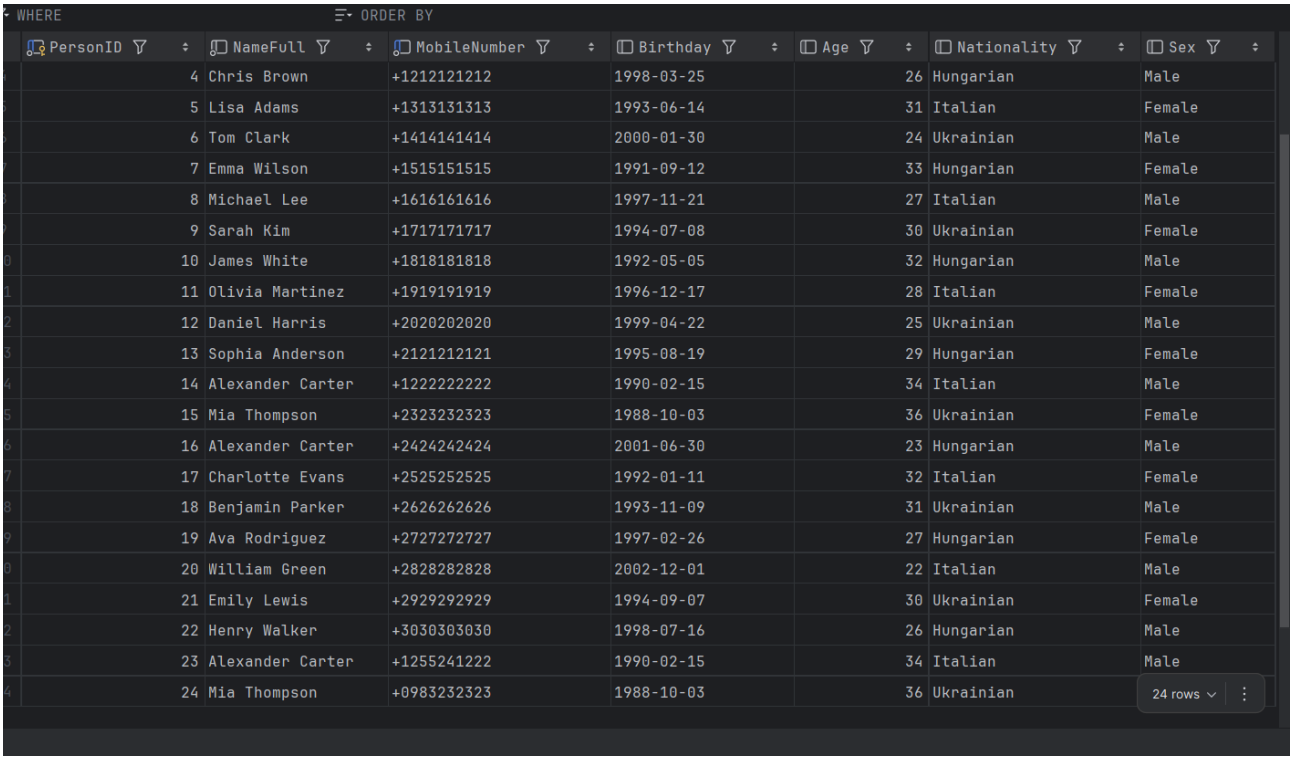


Output | soccer_cup.person | soccer_cup.selectedfemales

	PersonID	NameFull	MobileNumber	Birthday	Age	Nationality	Sex
1	7	Emma Wilson	+1515151515	1991-09-12	33	Hungarian	Female
2	13	Sophia Anderson	+2121212121	1995-08-19	29	Hungarian	Female
3	19	Ava Rodriguez	+2727272727	1997-02-26	27	Hungarian	Female
4	9	Sarah Kim	+1717171717	1994-07-08	30	Ukrainian	Female
5	15	Mia Thompson	+2323232323	1988-10-03	36	Ukrainian	Female
6	21	Emily Lewis	+2929292929	1994-09-07	30	Ukrainian	Female

Intersection (I want to return the set of persons who have the same NameFull and the same Birthday, but

Different Mobile numbers. They appear more than a 1 time in the table). To do that, I add some new rows with data to my csv to get a result.



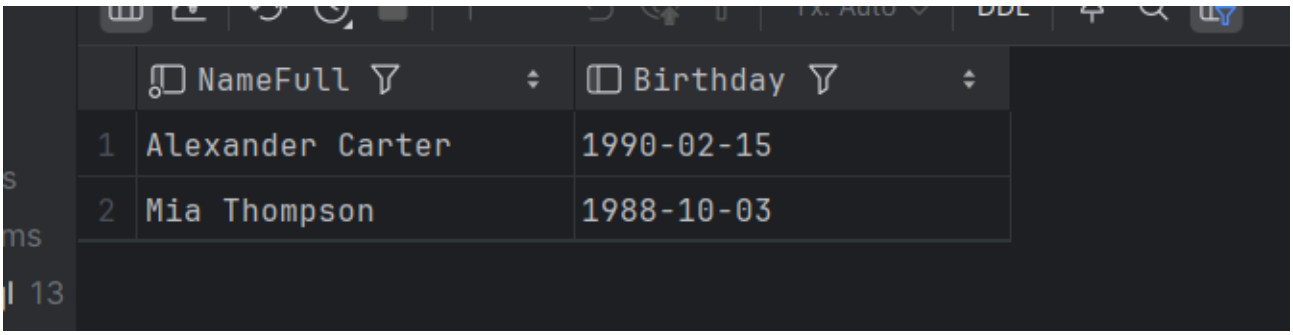
WHERE ORDER BY

PersonID	NameFull	MobileNumber	Birthday	Age	Nationality	Sex
4	Chris Brown	+1212121212	1998-03-25	26	Hungarian	Male
5	Lisa Adams	+1313131313	1993-06-14	31	Italian	Female
6	Tom Clark	+1414141414	2000-01-30	24	Ukrainian	Male
7	Emma Wilson	+1515151515	1991-09-12	33	Hungarian	Female
8	Michael Lee	+1616161616	1997-11-21	27	Italian	Male
9	Sarah Kim	+1717171717	1994-07-08	30	Ukrainian	Female
10	James White	+1818181818	1992-05-05	32	Hungarian	Male
11	Olivia Martinez	+1919191919	1996-12-17	28	Italian	Female
12	Daniel Harris	+2020202020	1999-04-22	25	Ukrainian	Male
13	Sophia Anderson	+2121212121	1995-08-19	29	Hungarian	Female
14	Alexander Carter	+1222222222	1990-02-15	34	Italian	Male
15	Mia Thompson	+2323232323	1988-10-03	36	Ukrainian	Female
16	Alexander Carter	+2424242424	2001-06-30	23	Hungarian	Male
17	Charlotte Evans	+2525252525	1992-01-11	32	Italian	Female
18	Benjamin Parker	+2626262626	1993-11-09	31	Ukrainian	Male
19	Ava Rodriguez	+2727272727	1997-02-26	27	Hungarian	Female
20	William Green	+2828282828	2002-12-01	22	Italian	Male
21	Emily Lewis	+2929292929	1994-09-07	30	Ukrainian	Female
22	Henry Walker	+3030303030	1998-07-16	26	Hungarian	Male
23	Alexander Carter	+1255241222	1990-02-15	34	Italian	Male
24	Mia Thompson	+0983232323	1988-10-03	36	Ukrainian	

24 rows

```sql
SELECT DISTINCT NameFull, Birthday
FROM Person
WHERE (NameFull, Birthday) IN (
    SELECT NameFull, Birthday
    FROM Person
    GROUP BY NameFull, Birthday
    HAVING COUNT(DISTINCT MobileNumber) > 1
);
```

Result (duplicates are not shown in output like in lab 7 example) :



	NameFull	Birthday
1	Alexander Carter	1990-02-15
2	Mia Thompson	1988-10-03

Difference:

If you want to find people whose mobile
numbers start with +1 and do not appear in the list of people whose
mobile numbers start with +2, you can use the following query

```
PersonID,NameFull,MobileNumber,Birthday,Age,Nationality,Sex
1,John Doe,+1234567890,1990-05-20,34,Hungarian,Male
2,Jane Smith,+0987654321,1985-07-15,38,Italian,Female
3,David Johnson,+1122334455,1995-09-10,29,Ukrainian,Male
4,Chris Brown,+1212121212,1998-03-25,26,Hungarian,Male
5,Lisa Adams,+1313131313,1993-06-14,31,Italian,Female
6,Tom Clark,+1414141414,2000-01-30,24,Ukrainian,Male
7,Emma Wilson,+1515151515,1991-09-12,33,Hungarian,Female
8,Michael Lee,+1616161616,1997-11-21,27,Italian,Male
9,Sarah Kim,+1717171717,1994-07-08,30,Ukrainian,Female
10,James White,+1818181818,1992-05-05,32,Hungarian,Male
11,Olivia Martinez,+1919191919,1996-12-17,28,Italian,Female
12,Daniel Harris,+2020202020,1999-04-22,25,Ukrainian,Male
13,Sophia Anderson,+2121212121,1995-08-19,29,Hungarian,Female
14,Alexander Carter,+1222222222,1990-02-15,34,Italian,Male
15,Mia Thompson,+2323232323,1988-10-03,36,Ukrainian,Female
16,Alexander Carter,+2424242424,2001-06-30,23,Hungarian,Male
17,Charlotte Evans,+2525252525,1992-01-11,32,Italian,Female
18,Benjamin Parker,+2626262626,1993-11-09,31,Ukrainian,Male
19,Ava Rodriguez,+2727272727,1997-02-26,27,Hungarian,Female
20,William Green,+2828282828,2002-12-01,22,Italian,Male
21,Emily Lewis,+2929292929,1994-09-07,30,Ukrainian,Female
22,Henry Walker,+3030303030,1998-07-16,26,Hungarian,Male
23,Alexander Carter,+1255241222,1990-02-15,34,Italian,Male
12,Daniel Harris,+2020202020,1999-04-22,25,Ukrainian,Male
24,Mia Thompson,+0983232323,1988-10-03,36,Ukrainian,Female
```

```sql
SELECT NameFull
FROM (
    SELECT NameFull, ANY_VALUE(SUBSTR(MobileNumber, 1, 2)) AS MobilePrefix
    FROM Person
    WHERE SUBSTR(MobileNumber, 1, 2) = '+1'
    GROUP BY NameFull
) AS Group1

EXCEPT

SELECT NameFull NameFull
FROM (
    SELECT NameFull, ANY_VALUE(SUBSTR(MobileNumber, 1, 2)) AS MobilePrefix
    FROM Person
    WHERE SUBSTR(MobileNumber, 1, 2) = '+2'
    GROUP BY NameFull
) AS Group2;
```

	NameFull
1	John Doe
2	David Johnson
3	Chris Brown
4	Lisa Adams
5	Tom Clark
6	Emma Wilson
7	Michael Lee
8	Sarah Kim
9	James White
10	Olivia Martinez

Selection:

If you want to get all players who has a position of ‘Forward’

Source table Player:

	yerID	JerseyNumber	Position	GoalsScored	CardsReceived	TotalMinut
1	1	10	Forward	5	1	
2	2	7	Midfielder	2	2	
3	3	5	Defender	0	3	
4	4	9	Forward	4	1	
5	5	8	Midfielder	1	0	
6	6	4	Defender	0	2	
7	7	3	Goalkeeper	0	1	
8	8	11	Forward	3	0	
9	9	6	Midfielder	2	2	
10	10	2	Defender	0	1	
11	11	1	Goalkeeper	0	0	
12	12	12	Forward	5	1	
13	13	14	Midfielder	2	2	
14	14	13	Defender	0	3	

Selection Query:

```
#Selection from Player:
SELECT * FROM player WHERE Position='Forward';
```

The result is all Forward players:

Output | soccer_cup.player | Result 134-2

	PlayerID	JerseyNumber	Position	GoalsScored	CardsReceived	TotalMinutesPlayed	TeamID
1	1	10	Forward	5	1	270	1
2	4	9	Forward	4	1	260	1
3	8	11	Forward	3	0	255	2
4	12	12	Forward	5	1	265	2
5	15	16	Forward	4	1	260	3
6	19	20	Forward	3	0	255	3

Projection query:

Source table Person. Here Some similar persons: Alexander Carter with the same Nationality.

WHERE ORDER BY

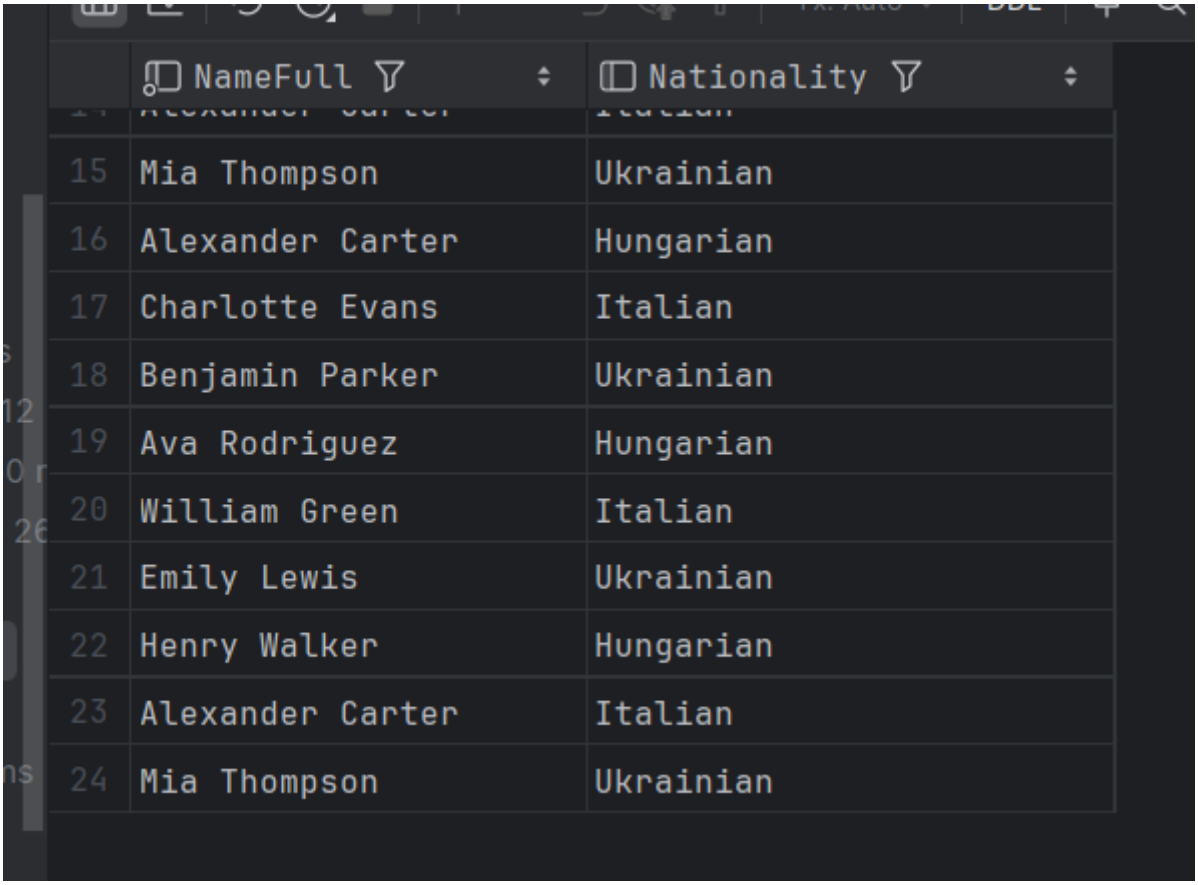
	PersonID	NameFull	MobileNumber	Birthday	Age	Nationality
1	1	John Doe	+1234567890	1990-05-20	34	Hungarian
2	2	Jane Smith	+0987654321	1985-07-15	38	Italian
3	3	David Johnson	+1122334455	1995-09-10	29	Ukrainian
4	4	Chris Brown	+1212121212	1998-03-25	26	Hungarian
5	5	Lisa Adams	+1313131313	1993-06-14	31	Italian
6	6	Tom Clark	+1414141414	2000-01-30	24	Ukrainian
7	7	Emma Wilson	+1515151515	1991-09-12	33	Hungarian
8	8	Michael Lee	+1616161616	1997-11-21	27	Italian
9	9	Sarah Kim	+1717171717	1994-07-08	30	Ukrainian
10	10	James White	+1818181818	1992-05-05	32	Hungarian
11	11	Olivia Martinez	+1919191919	1996-12-17	28	Italian
12	12	Daniel Harris	+2020202020	1999-04-22	25	Ukrainian
13	13	Sophia Anderson	+2121212121	1995-08-19	29	Hungarian
14	14	Alexander Carter	+1222222222	1990-02-15	34	Italian
15	15	Mia Thompson	+2323232323	1988-10-03	36	Ukrainian
16	16	Alexander Carter	+2424242424	2001-06-30	23	Hungarian
17	17	Charlotte Evans	+2525252525	1992-01-11	32	Italian
18	18	Benjamin Parker	+2626262626	1993-11-09	31	Ukrainian
19	19	Ava Rodriguez	+2727272727	1997-02-26	27	Hungarian
20	20	William Green	+2828282828	2002-12-01	22	Italian
21	21	Emily Lewis	+2929292929	1994-09-07	30	Ukrai

24 rows

Projection query and Projection Distinct Query:

```
59
60      # Projection:
61      SELECT NameFull, Nationality FROM person;
62      # Distinct Projection:
63      SELECT DISTINCT  NameFull, Nationality FROM person;
```

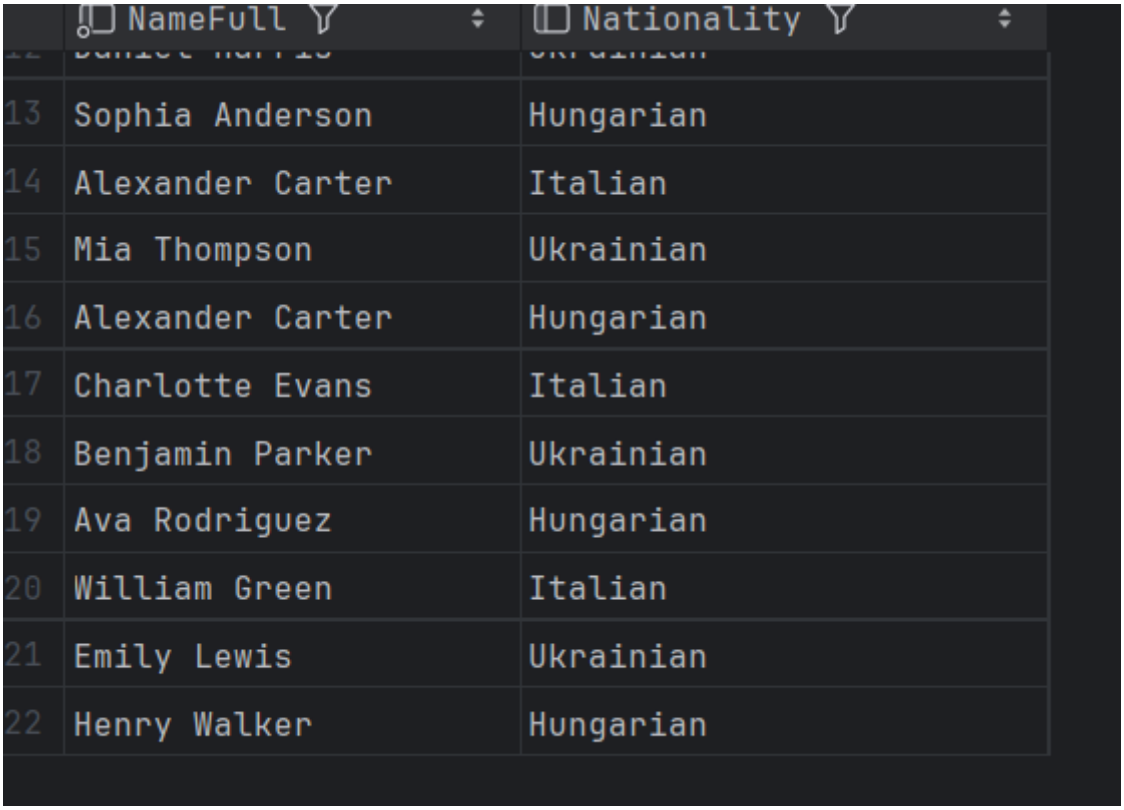
Projection query result. There are repetitions:



	NameFull	Nationality
15	Mia Thompson	Ukrainian
16	Alexander Carter	Hungarian
17	Charlotte Evans	Italian
18	Benjamin Parker	Ukrainian
19	Ava Rodriguez	Hungarian
20	William Green	Italian
21	Emily Lewis	Ukrainian
22	Henry Walker	Hungarian
23	Alexander Carter	Italian
24	Mia Thompson	Ukrainian

As you can see here 24 persons.

If I run projection distinct query, then duplicates will be retrived:

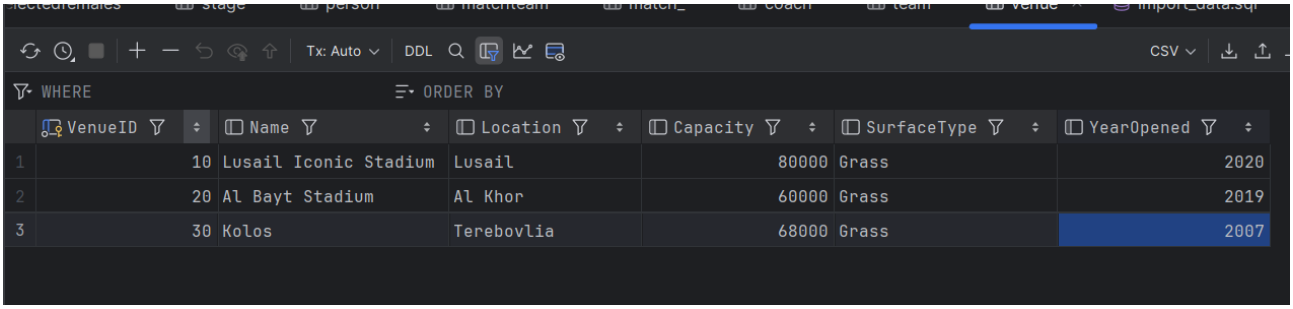


	NameFull	Nationality
13	Sophia Anderson	Hungarian
14	Alexander Carter	Italian
15	Mia Thompson	Ukrainian
16	Alexander Carter	Hungarian
17	Charlotte Evans	Italian
18	Benjamin Parker	Ukrainian
19	Ava Rodriguez	Hungarian
20	William Green	Italian
21	Emily Lewis	Ukrainian
22	Henry Walker	Hungarian

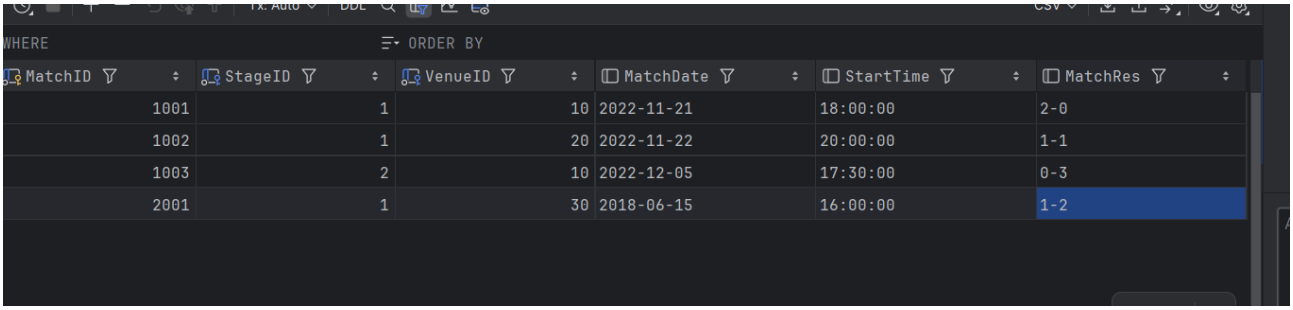
Here is 22. It means, that duplicates such as Alexander Carter, Italian are removed.

Cartesian Product:

Source data:

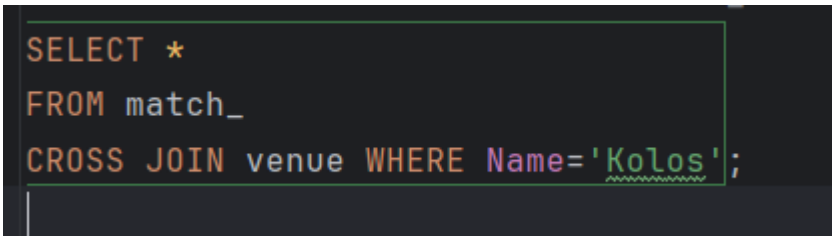


	VenueID	Name	Location	Capacity	SurfaceType	YearOpened
1	10	Lusail Iconic Stadium	Lusail	80000	Grass	2020
2	20	Al Bayt Stadium	Al Khor	60000	Grass	2019
3	30	Kolos	Terebovlia	68000	Grass	2007



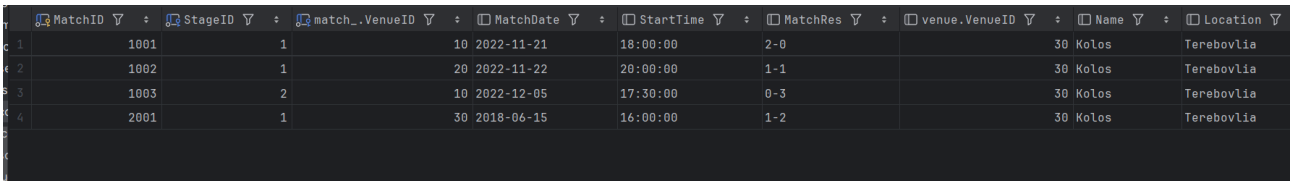
MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1001	1	10	2022-11-21	18:00:00	2-0
1002	1	20	2022-11-22	20:00:00	1-1
1003	2	10	2022-12-05	17:30:00	0-3
2001	1	30	2018-06-15	16:00:00	1-2

Cartesian product Query:



```
SELECT *
FROM match_
CROSS JOIN venue WHERE Name='Kolos';
```

Result is cartesian product of 2 tables. Matches which are provided on Kolos Venue



	MatchID	StageID	match_.VenueID	MatchDate	StartTime	MatchRes	venue.VenueID	Name	Location
1	1001	1	10	2022-11-21	18:00:00	2-0	30	Kolos	Terebovlia
2	1002	1	20	2022-11-22	20:00:00	1-1	30	Kolos	Terebovlia
3	1003	2	10	2022-12-05	17:30:00	0-3	30	Kolos	Terebovlia
4	2001	1	30	2018-06-15	16:00:00	1-2	30	Kolos	Terebovlia

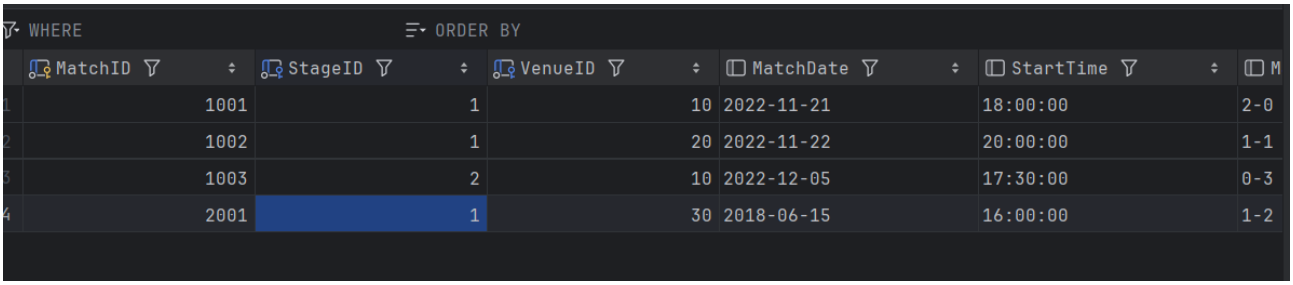
Step 5:

Update query

At the input, I have created tables which are populated in step 3 with the data obtained from csv. There is a lot of data which I can test. You can see source data (input) below in examples

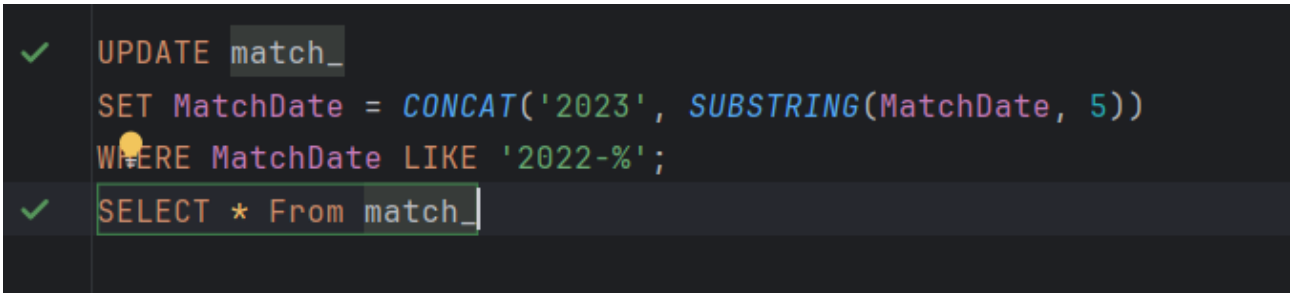
I used the following order to show you work: Source data -> query -> Result

Source table, where the year of first 3 matches is 2022. I want to change it to 2023



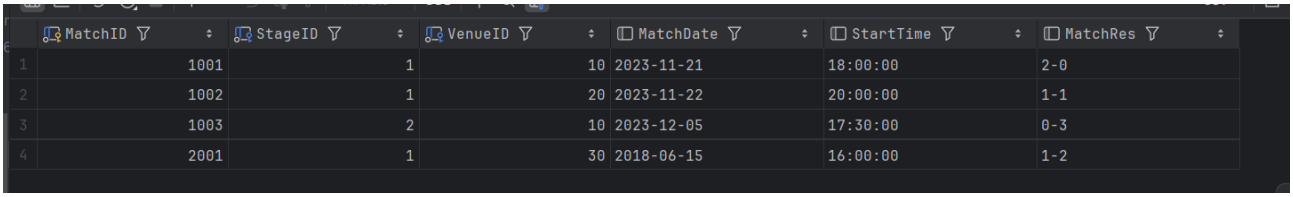
	MatchID	StageID	VenueID	MatchDate	StartTime	M
1	1001	1	10	2022-11-21	18:00:00	2-0
2	1002	1	20	2022-11-22	20:00:00	1-1
3	1003	2	10	2022-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

My query:



```
UPDATE match_
SET MatchDate = CONCAT('2023', SUBSTRING(MatchDate, 5))
WHERE MatchDate LIKE '2022-%';
SELECT * From match_
```

My result:



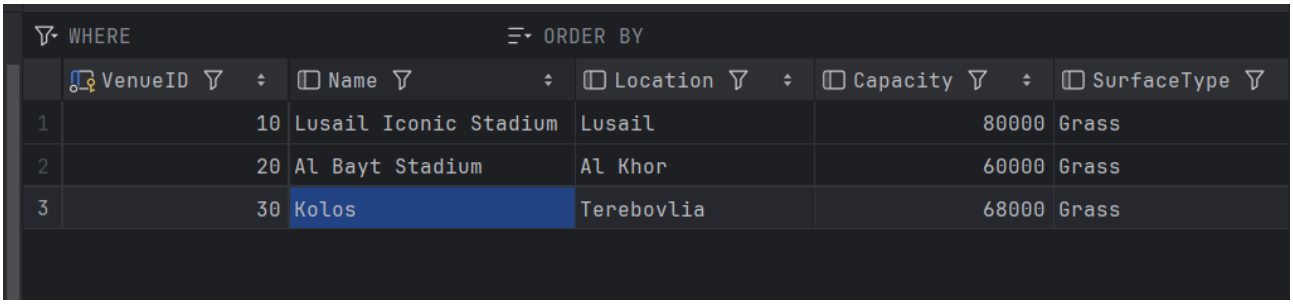
	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	10	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003	2	10	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

Cascad Update:

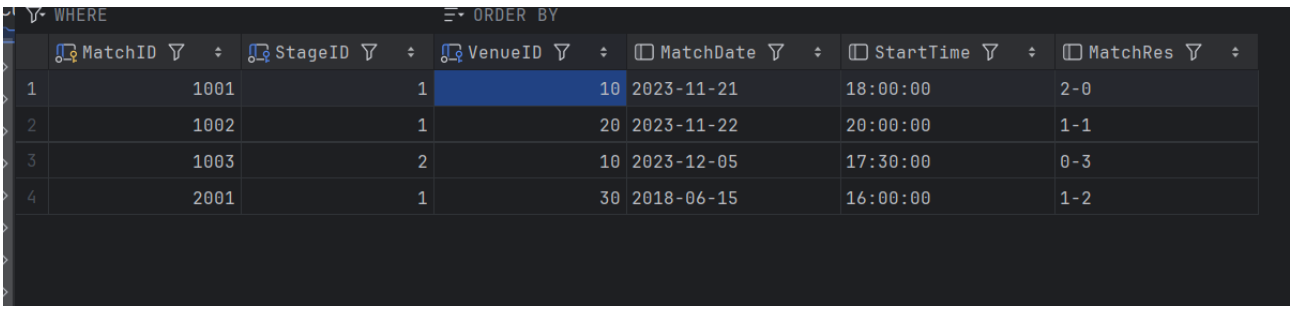
Match_ has a foreign key referencing to VenueID in Venue.

So, If I use a cascade update to update the value of VenueID in Venue, it should also change the value of VenueID in Match_

Source tables (Venue and Match_):



	VenueID	Name	Location	Capacity	SurfaceType
1	10	Lusail Iconic Stadium	Lusail	80000	Grass
2	20	Al Bayt Stadium	Al Khor	60000	Grass
3	30	Kolos	Terebovlia	68000	Grass



	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	10	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003	2	10	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

The queris to change the value of VenueID, e.g. let we change 10 to 150.

```sql
#Cascade update Match_
SET FOREIGN_KEY_CHECKS = 0;

ALTER TABLE Match_ DROP FOREIGN KEY match__ibfk_1;
ALTER TABLE Match_ DROP FOREIGN KEY match__ibfk_2;

ALTER TABLE Match_
ADD CONSTRAINT match__ibfk_1 FOREIGN KEY (`StageID`)
REFERENCES Stage (`StageID`)
ON UPDATE CASCADE;

ALTER TABLE Match_
ADD CONSTRAINT match__ibfk_2 FOREIGN KEY (`VenueID`)
REFERENCES Venue (`VenueID`)
ON UPDATE CASCADE;

SET FOREIGN_KEY_CHECKS = 1;

UPDATE Venue
SET VenueID = 150
WHERE VenueID = 10;
```

The results. As we can see, the VenueID was changed not only in Venue but also Match_, where it is a foreign key:

	VenueID	Name	Location	Capacity	SurfaceType	YearOpened
1	20	Al Bayt Stadium	Al Khor	60000	Grass	2019
2	30	Kolos	Terebovlia	68000	Grass	2007
3	150	Lusail Iconic Stadium	Lusail	80000	Grass	2020

WHERE ORDER BY

	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	150	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003	2	150	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

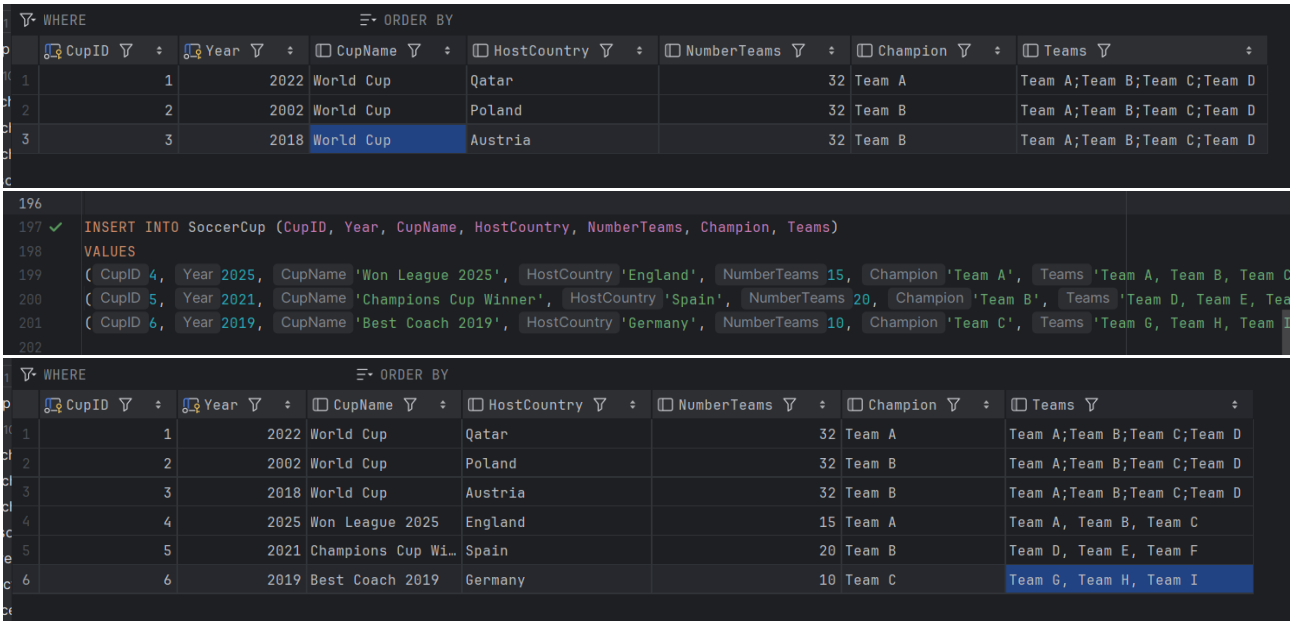
Step 6: Add new data

Input: the tables with data, populated in step 3. Beneath I provided the examples where you can see input data table.

I used the following order to show you work: Source data -> query -> Result

Insert:

Lets add some more tournaments to soccer cup table:

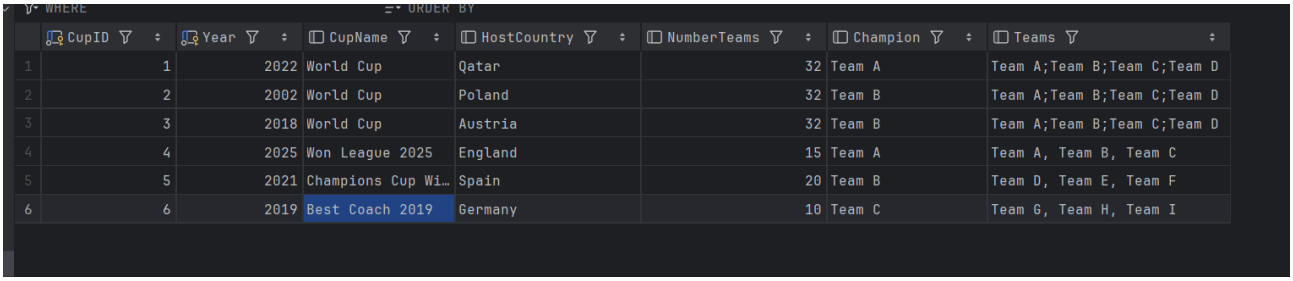


	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D

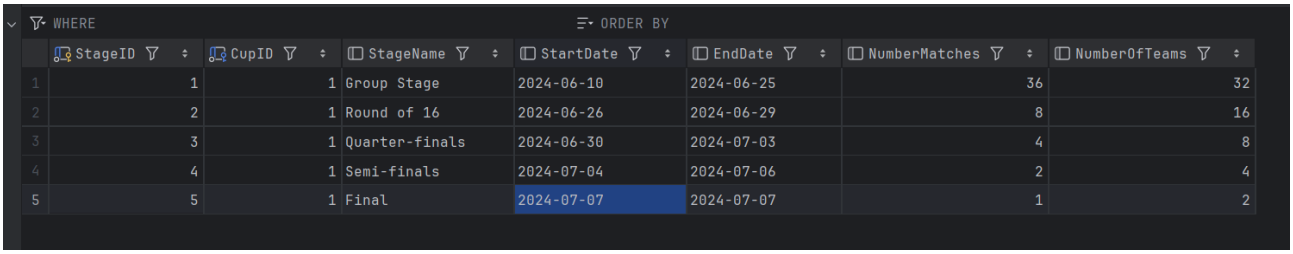
```
196
197 ✓ INSERT INTO SoccerCup (CupID, Year, CupName, HostCountry, NumberTeams, Champion, Teams)
198   VALUES
199   ( CupID 4, Year 2025, CupName 'Won League 2025', HostCountry 'England', NumberTeams 15, Champion 'Team A', Teams 'Team A, Team B, Team C
200   ( CupID 5, Year 2021, CupName 'Champions Cup Winner', HostCountry 'Spain', NumberTeams 20, Champion 'Team B', Teams 'Team D, Team E, Tea
201   ( CupID 6, Year 2019, CupName 'Best Coach 2019', HostCountry 'Germany', NumberTeams 10, Champion 'Team C', Teams 'Team G, Team H, Team I
202
```

	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D
4	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
5	5	2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
6	6	2019	Best Coach 2019	Germany	10	Team C	Team G, Team H, Team I

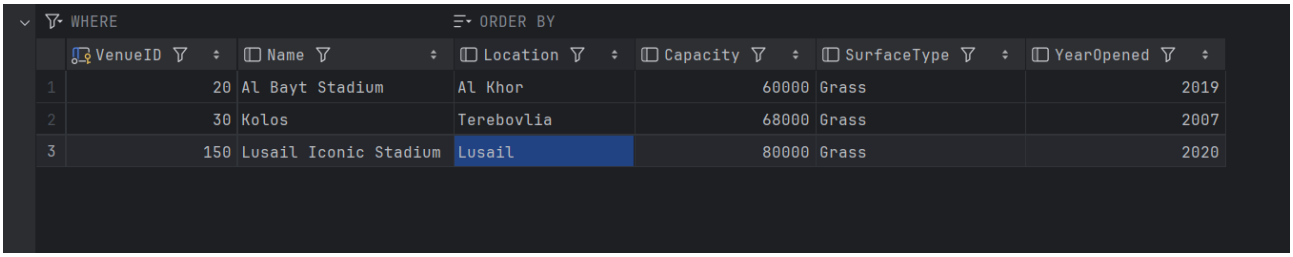
Complex insert:



	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D
4	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
5	5	2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
6	6	2019	Best Coach 2019	Germany	10	Team C	Team G, Team H, Team I



	StageID	CupID	StageName	StartDate	EndDate	NumberMatches	NumberOfTeams
1	1	1	Group Stage	2024-06-10	2024-06-25	36	32
2	2	1	Round of 16	2024-06-26	2024-06-29	8	16
3	3	1	Quarter-finals	2024-06-30	2024-07-03	4	8
4	4	1	Semi-finals	2024-07-04	2024-07-06	2	4
5	5	1	Final	2024-07-07	2024-07-07	1	2



	VenueID	Name	Location	Capacity	SurfaceType	YearOpened
1	20	Al Bayt Stadium	Al Khor	60000	Grass	2019
2	30	Kolos	Terebovlia	68000	Grass	2007
3	150	Lusail Iconic Stadium	Lusail	80000	Grass	2020

WHERE ORDER BY

	TeamID	TeamName	HomeCountry	GoalsTotal	PointsTotal
1	1	Team A	Country A	45	20
2	2	Team B	Country B	50	22
3	3	Team C	Country C	37	18
4	4	Team D	Country D	40	19

WHERE ORDER BY

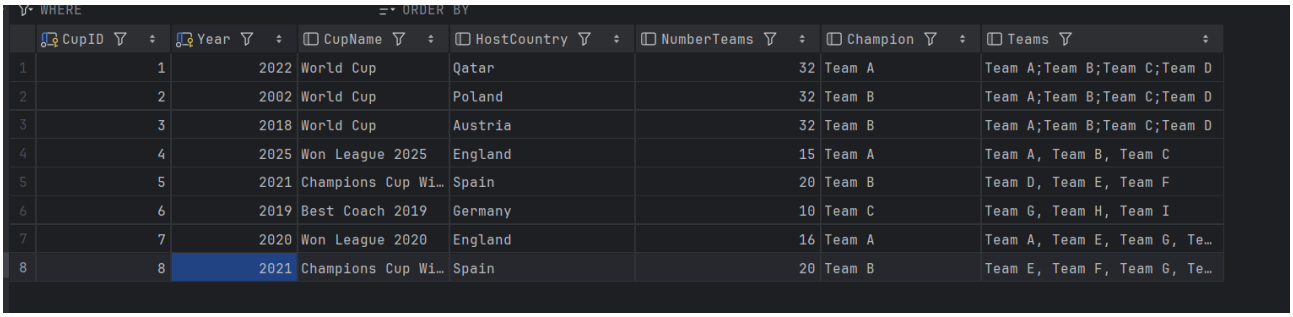
	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	150	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003		150	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2

Complex insert queries:

```sql
# complex insertion query that adds new data to several related tables, including the
# foreign key values
-- Insert Soccer Cups
INSERT INTO SoccerCup (CupID, Year, CupName, HostCountry, NumberTeams, Champion, Teams)
VALUES
( CupID 7, Year 2020, CupName 'Won League 2020', HostCountry 'England', NumberTeams 16, Champion 'Team A', Teams 'Team A, Team E, Team G, Team D'),
( CupID 8, Year 2021, CupName 'Champions Cup Winner', HostCountry 'Spain', NumberTeams 20, Champion 'Team B', Teams 'Team E, Team F, Team G, Team H');

-- Insert Stages (linked to SoccerCup)
INSERT INTO Stage (StageID, CupID, StageName, StartDate, EndDate, NumberMatches, NumberOfTeams)
VALUES
( StageID 6, CupID 2, StageName 'Group Stage', StartDate '2020-06-01', EndDate '2020-06-15', NumberMatches 8, NumberOfTeams 16),
( StageID 7, CupID 2, StageName 'Final', StartDate '2020-07-01', EndDate '2020-07-05', NumberMatches 1, NumberOfTeams 2),
( StageID 8, CupID 3, StageName 'Semifinal', StartDate '2021-07-10', EndDate '2021-07-15', NumberMatches 2, NumberOfTeams 4),
( StageID 9, CupID 3, StageName 'Final', StartDate '2021-07-20', EndDate '2021-07-25', NumberMatches 1, NumberOfTeams 2);

-- Insert Venues
INSERT INTO Venue (VenueID, Name, Location, Capacity, SurfaceType, YearOpened)
VALUES
( VenueID 1, Name 'Wembley Stadium', Location 'London, England', Capacity 90000, SurfaceType 'Grass', YearOpened 1923),
( VenueID 2, Name 'Camp Nou', Location 'Barcelona, Spain', Capacity 99354, SurfaceType 'Grass', YearOpened 1957),
( VenueID 3, Name 'Allianz Arena', Location 'Munich, Germany', Capacity 75000, SurfaceType 'Hybrid Grass', YearOpened 2005);

-- Insert Teams
INSERT INTO Team (TeamID, TeamName, HomeCountry, GoalsTotal, PointsTotal) VALUES
( TeamID 5, TeamName 'Team E', HomeCountry 'England', GoalsTotal 45, PointsTotal 90),
( TeamID 6, TeamName 'Team F', HomeCountry 'Spain', GoalsTotal 38, PointsTotal 85),
```

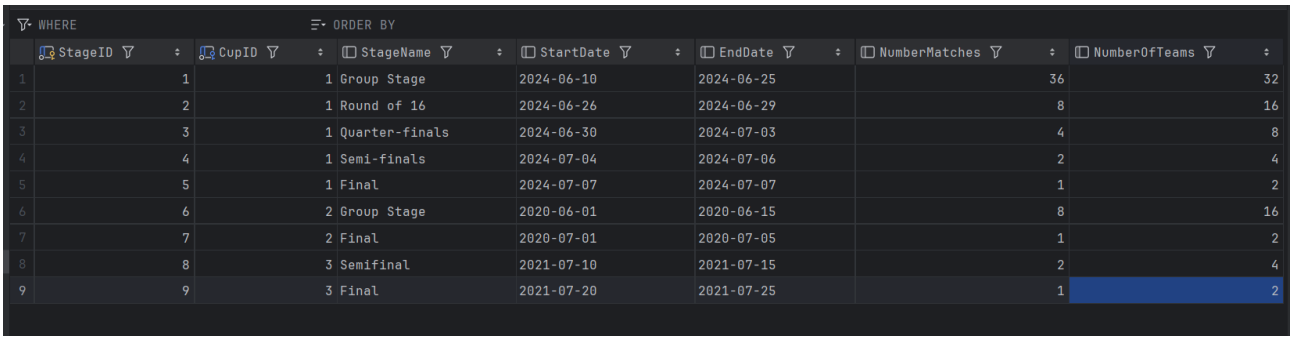
```sql
-- Insert Venues
INSERT INTO Venue (VenueID, Name, Location, Capacity, SurfaceType, YearOpened)
VALUES
( VenueID 1, Name 'Wembley Stadium', Location 'London, England', Capacity 90000, SurfaceType 'Grass', YearOpened 1923),
( VenueID 2, Name 'Camp Nou', Location 'Barcelona, Spain', Capacity 99354, SurfaceType 'Grass', YearOpened 1957),
( VenueID 3, Name 'Allianz Arena', Location 'Munich, Germany', Capacity 75000, SurfaceType 'Hybrid Grass', YearOpened 2005);

-- Insert Teams
INSERT INTO Team (TeamID, TeamName, HomeCountry, GoalsTotal, PointsTotal) VALUES
( TeamID 5, TeamName 'Team E', HomeCountry 'England', GoalsTotal 45, PointsTotal 90),
( TeamID 6, TeamName 'Team F', HomeCountry 'Spain', GoalsTotal 38, PointsTotal 85),
( TeamID 7, TeamName 'Team G', HomeCountry 'Germany', GoalsTotal 50, PointsTotal 92),
( TeamID 8, TeamName 'Team H', HomeCountry 'France', GoalsTotal 42, PointsTotal 88);

-- Insert Matches (linked to Stage, Venue)
INSERT INTO Match_ (MatchID, StageID, VenueID, MatchDate, StartTime, MatchRes)
VALUES
( MatchID 3001, StageID 1, VenueID 1, MatchDate '2020-06-10', StartTime '18:00:00', MatchRes '2-1'),
( MatchID 3002, StageID 2, VenueID 2, MatchDate '2020-07-05', StartTime '20:30:00', MatchRes '3-0'),
( MatchID 3003, StageID 3, VenueID 3, MatchDate '2021-07-12', StartTime '21:00:00', MatchRes '1-1'),
( MatchID 3004, StageID 4, VenueID 1, MatchDate '2021-07-25', StartTime '22:00:00', MatchRes '2-1');
```

Results:



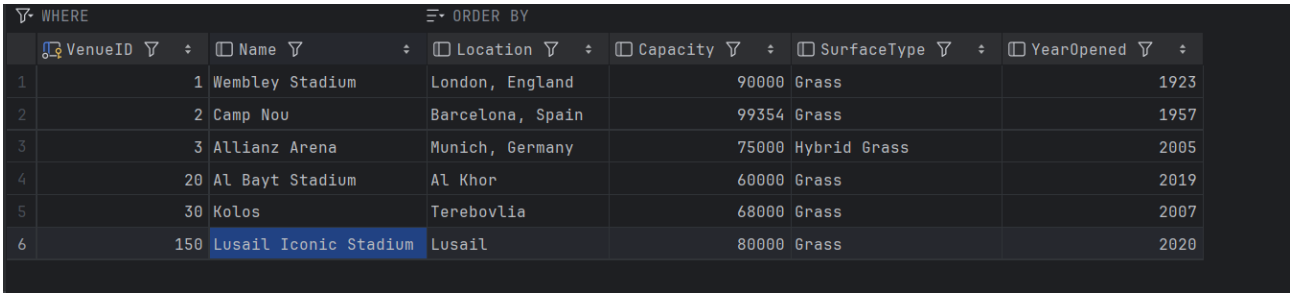
WHERE ORDER BY

	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D
4	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
5	5	2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
6	6	2019	Best Coach 2019	Germany	10	Team C	Team G, Team H, Team I
7	7	2020	Won League 2020	England	16	Team A	Team A, Team E, Team G, Te…
8	8	2021	Champions Cup Wi…	Spain	20	Team B	Team E, Team F, Team G, Te…



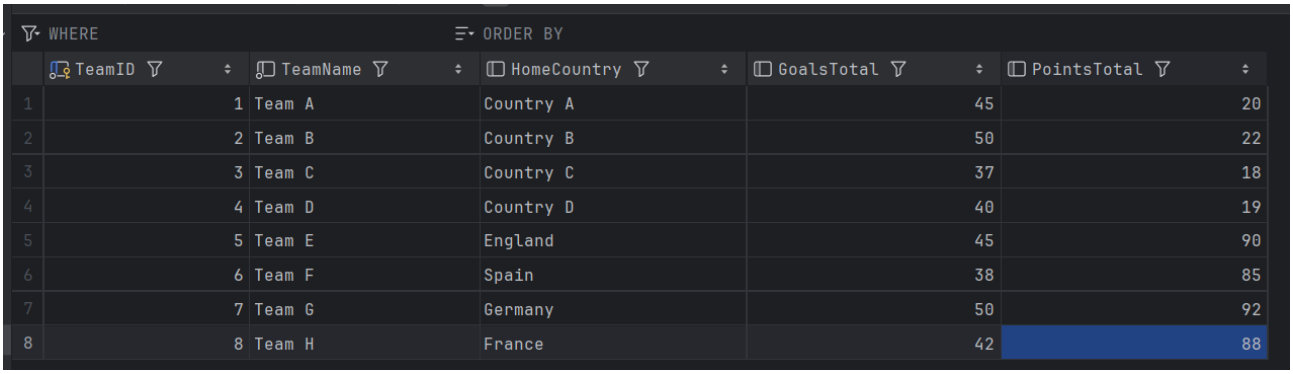
WHERE ORDER BY

	StageID	CupID	StageName	StartDate	EndDate	NumberMatches	NumberOfTeams
1	1	1	Group Stage	2024-06-10	2024-06-25	36	32
2	2	1	Round of 16	2024-06-26	2024-06-29	8	16
3	3	1	Quarter-finals	2024-06-30	2024-07-03	4	8
4	4	1	Semi-finals	2024-07-04	2024-07-06	2	4
5	5	1	Final	2024-07-07	2024-07-07	1	2
6	6	2	Group Stage	2020-06-01	2020-06-15	8	16
7	7	2	Final	2020-07-01	2020-07-05	1	2
8	8	3	Semifinal	2021-07-10	2021-07-15	2	4
9	9	3	Final	2021-07-20	2021-07-25	1	2



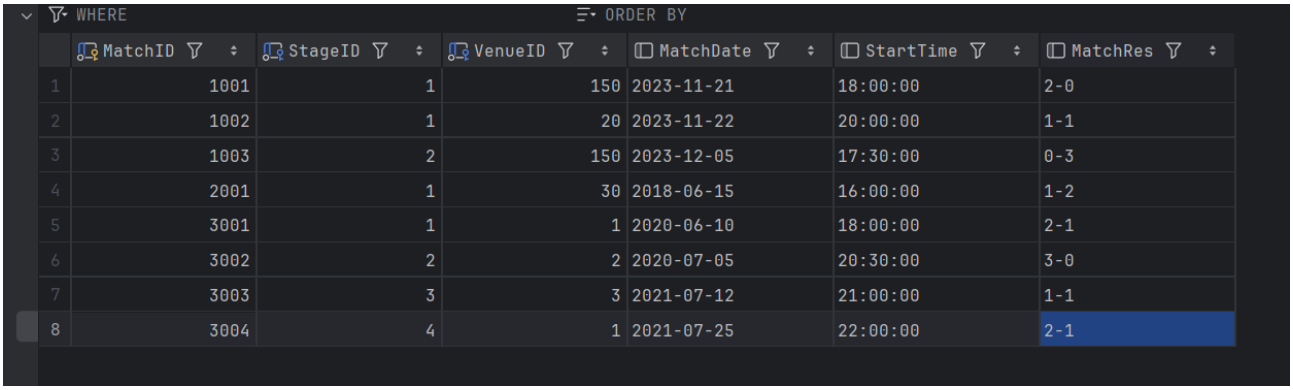
WHERE ORDER BY

	VenueID	Name	Location	Capacity	SurfaceType	YearOpened
1	1	Wembley Stadium	London, England	90000	Grass	1923
2	2	Camp Nou	Barcelona, Spain	99354	Grass	1957
3	3	Allianz Arena	Munich, Germany	75000	Hybrid Grass	2005
4	20	Al Bayt Stadium	Al Khor	60000	Grass	2019
5	30	Kolos	Terebovlia	68000	Grass	2007
6	150	Lusail Iconic Stadium	Lusail	80000	Grass	2020



WHERE ORDER BY

	TeamID	TeamName	HomeCountry	GoalsTotal	PointsTotal
1	1	Team A	Country A	45	20
2	2	Team B	Country B	50	22
3	3	Team C	Country C	37	18
4	4	Team D	Country D	40	19
5	5	Team E	England	45	90
6	6	Team F	Spain	38	85
7	7	Team G	Germany	50	92
8	8	Team H	France	42	88

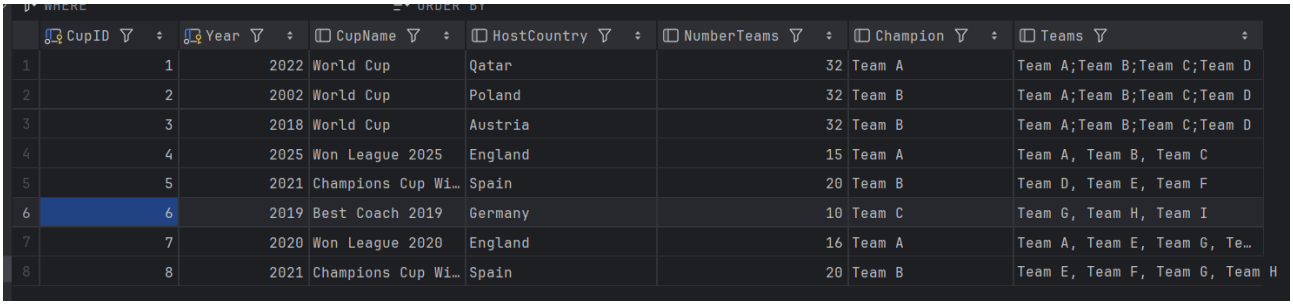


WHERE ORDER BY

	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	150	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003	2	150	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2
5	3001	1	1	2020-06-10	18:00:00	2-1
6	3002	2	2	2020-07-05	20:30:00	3-0
7	3003	3	3	2021-07-12	21:00:00	1-1
8	3004	4	1	2021-07-25	22:00:00	2-1

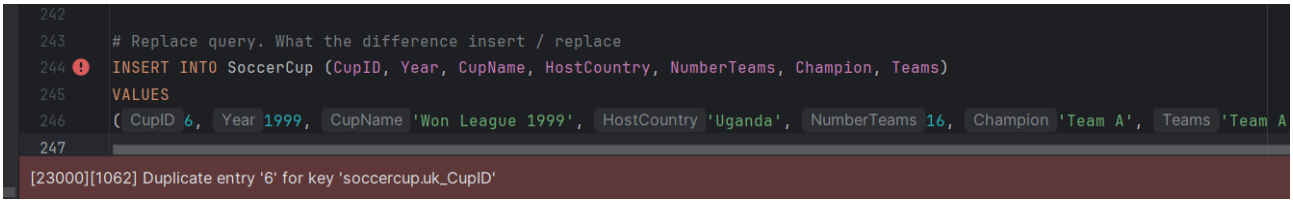
Replace:

A REPLACE query in MySQL is used to insert data into a table while replacing an existing row if there is a duplicate primary key or unique key conflict. It works like INSERT, but if a row with the same primary key already exists, it deletes the old row and inserts the new one.



	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D
4	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
5	5	2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
6	6	2019	Best Coach 2019	Germany	10	Team C	Team G, Team H, Team I
7	7	2020	Won League 2020	England	16	Team A	Team A, Team E, Team G, Te…
8	8	2021	Champions Cup Wi…	Spain	20	Team B	Team E, Team F, Team G, Team H

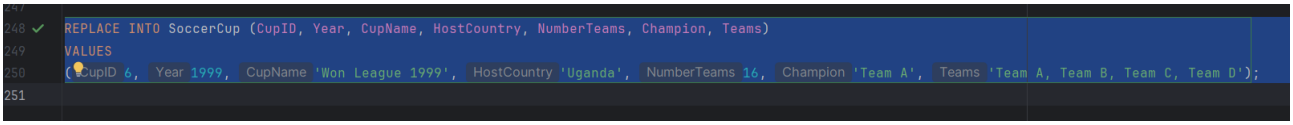
Try insert and result:



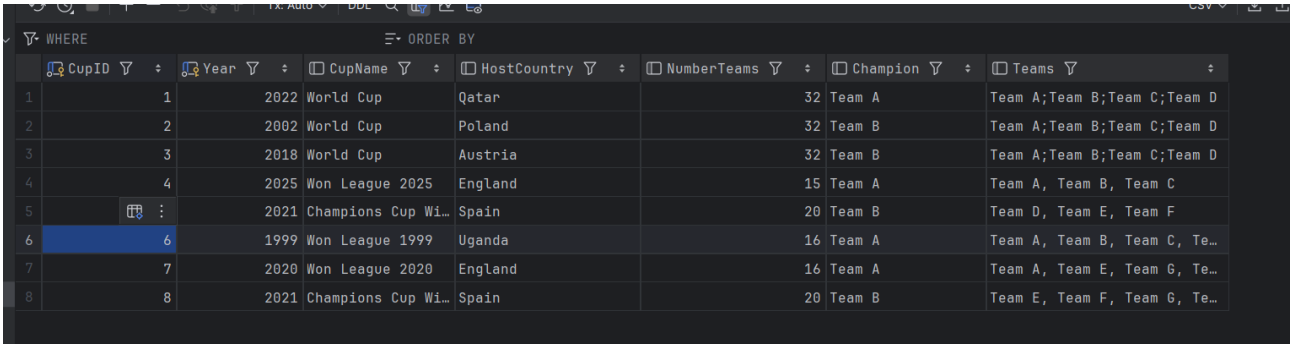
```
242
243  # Replace query. What the difference insert / replace
244  INSERT INTO SoccerCup (CupID, Year, CupName, HostCountry, NumberTeams, Champion, Teams)
245  VALUES
246  ( CupID 6, Year 1999, CupName 'Won League 1999', HostCountry 'Uganda', NumberTeams 16, Champion 'Team A', Teams 'Team A
247
```

[23000][1062] Duplicate entry '6' for key 'soccercup.uk_CupID'

Try replace and result:



```
248  REPLACE INTO SoccerCup (CupID, Year, CupName, HostCountry, NumberTeams, Champion, Teams)
249  VALUES
250  ( CupID 6, Year 1999, CupName 'Won League 1999', HostCountry 'Uganda', NumberTeams 16, Champion 'Team A', Teams 'Team A, Team B, Team C, Team D');
251
```



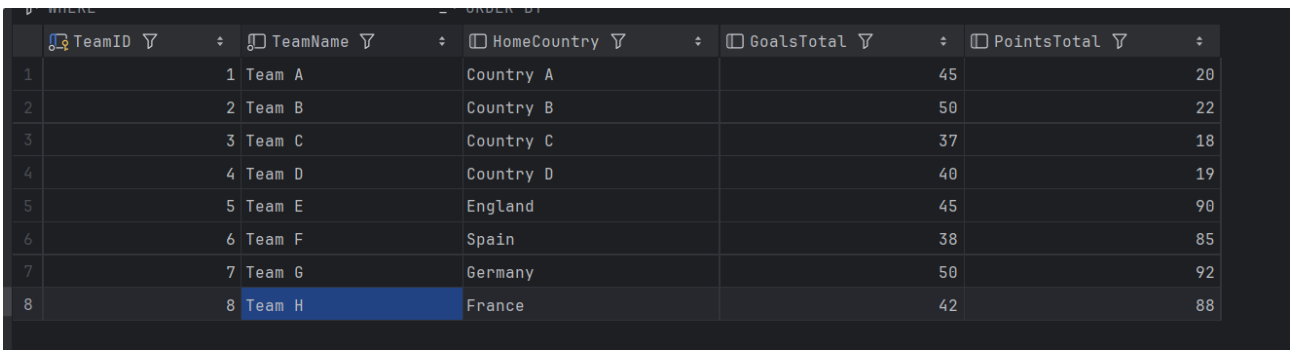
WHERE ORDER BY

	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D
2	2	2002	World Cup	Poland	32	Team B	Team A;Team B;Team C;Team D
3	3	2018	World Cup	Austria	32	Team B	Team A;Team B;Team C;Team D
4	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
5		2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
6	6	1999	Won League 1999	Uganda	16	Team A	Team A, Team B, Team C, Te…
7	7	2020	Won League 2020	England	16	Team A	Team A, Team E, Team G, Te…
8	8	2021	Champions Cup Wi…	Spain	20	Team B	Team E, Team F, Team G, Te…

As you can see, all works with replace query

Delete

Lets do a simple delete from Team. Lets delete Team H

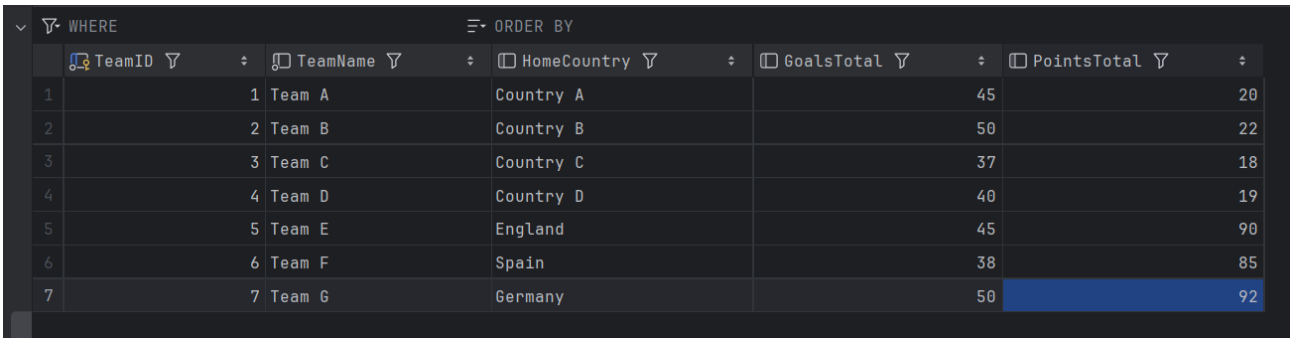


	TeamID	TeamName	HomeCountry	GoalsTotal	PointsTotal
1	1	Team A	Country A	45	20
2	2	Team B	Country B	50	22
3	3	Team C	Country C	37	18
4	4	Team D	Country D	40	19
5	5	Team E	England	45	90
6	6	Team F	Spain	38	85
7	7	Team G	Germany	50	92
8	8	Team H	France	42	88

Delete Query:

```
# Delete query
DELETE FROM Team
WHERE TeamName = 'Team H'
```

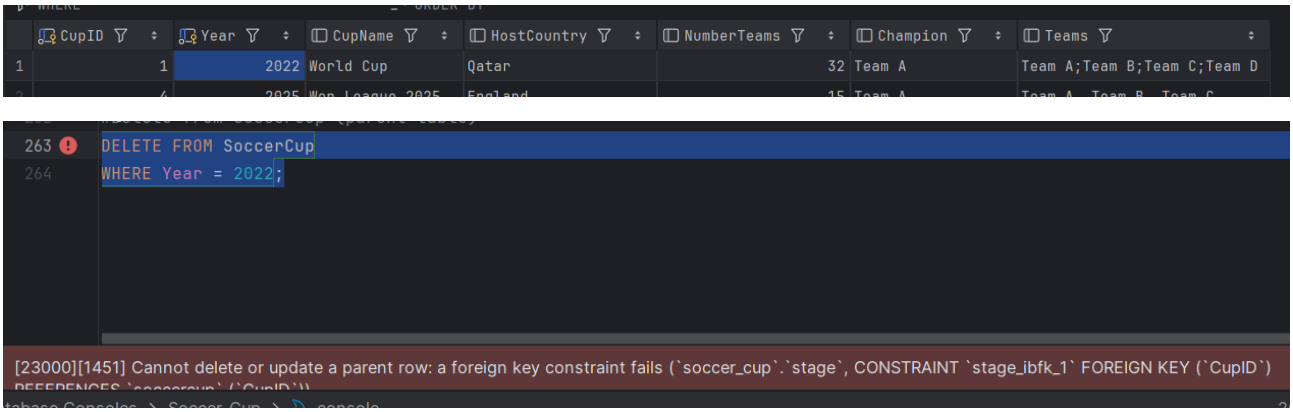
Results. You can see that Team H was deleted



TeamID	TeamName	HomeCountry	GoalsTotal	PointsTotal
1	Team A	Country A	45	20
2	Team B	Country B	50	22
3	Team C	Country C	37	18
4	Team D	Country D	40	19
5	Team E	England	45	90
6	Team F	Spain	38	85
7	Team G	Germany	50	92

Lets do a complex delete. To delete row with a data used in other tables you should firstly delete this data from child tables (tables, which use the data as foreign keys)

Here is an example with an error

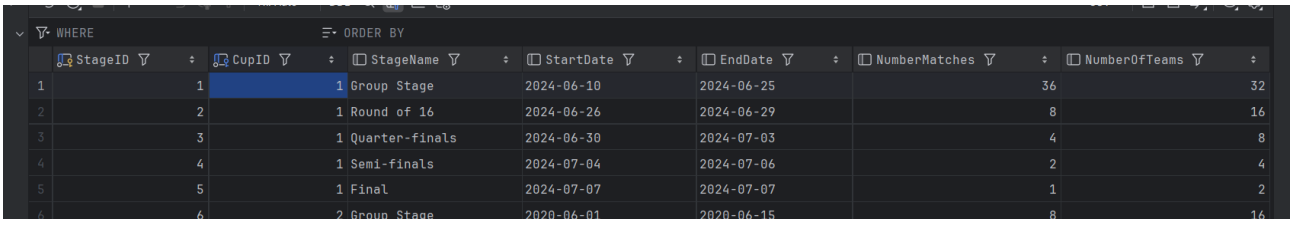


CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	2022	World Cup	Qatar	32	Team A	Team A;Team B;Team C;Team D

```
263 DELETE FROM SoccerCup
264 WHERE Year = 2022;
```

[23000][1451] Cannot delete or update a parent row: a foreign key constraint fails (`soccer_cup`.`stage`, CONSTRAINT `stage_ibfk_1` FOREIGN KEY (`CupID`)

Table related to soccer cup table (CupID = 1) in Stage:



StageID	CupID	StageName	StartDate	EndDate	NumberMatches	NumberOfTeams
1	1	Group Stage	2024-06-10	2024-06-25	36	32
2	1	Round of 16	2024-06-26	2024-06-29	8	16
3	1	Quarter-finals	2024-06-30	2024-07-03	4	8
4	1	Semi-finals	2024-07-04	2024-07-06	2	4
5	1	Final	2024-07-07	2024-07-07	1	2
6	2	Group Stage	2020-06-01	2020-06-15	8	16

WHERE ORDER BY

	MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes
1	1001	1	150	2023-11-21	18:00:00	2-0
2	1002	1	20	2023-11-22	20:00:00	1-1
3	1003	2	150	2023-12-05	17:30:00	0-3
4	2001	1	30	2018-06-15	16:00:00	1-2
5	3001	1	1	2020-06-10	18:00:00	2-1
6	3002	2	2	2020-07-05	20:30:00	3-0
7	3003	3	3	2021-07-12	21:00:00	1-1
8	3004	4	1	2021-07-25	22:00:00	2-1

Tx: Auto DDL

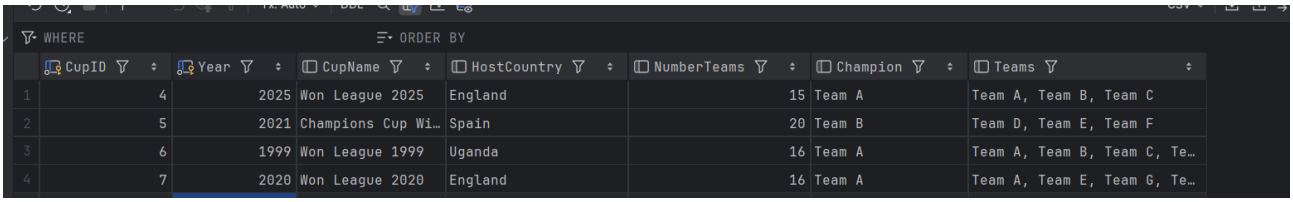
WHERE ORDER BY

	MatchID	TeamID	GoalsScored	IsHomeTeam
1	1001	1	2	1
2	1001	2	0	0
3	1002	3	1	1
4	1002	4	1	0
5	1003	2	0	1
6	1003	3	3	0
7	2001	1	1	1
8	2001	4	2	0

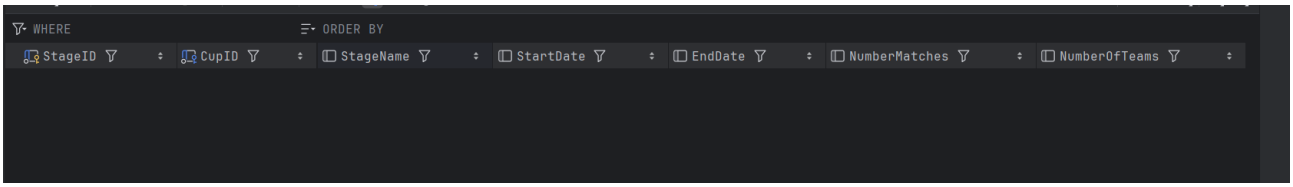
The cascade of queries to delete:

```sql
#Delete from matchteam (dependent on Match_)
DELETE FROM matchteam
WHERE MatchID IN (
    SELECT MatchID
    FROM Match_
    WHERE StageID IN (
        SELECT StageID
        FROM Stage
        WHERE CupID = (SELECT CupID FROM SoccerCup WHERE Year = 2022)
    )
);
# # Delete from Stage (dependent on SoccerCup)
DELETE FROM Match_
WHERE StageID IN (
    SELECT StageID
    FROM Stage
    WHERE CupID = (SELECT CupID FROM SoccerCup WHERE Year = 2022)
);

DELETE FROM Stage
WHERE CupID = (SELECT CupID FROM SoccerCup WHERE Year = 2022);
#Delete from SoccerCup (parent table)
DELETE FROM SoccerCup
WHERE Year = 2022;
```

And results in the connected tables:



	CupID	Year	CupName	HostCountry	NumberTeams	Champion	Teams
1	4	2025	Won League 2025	England	15	Team A	Team A, Team B, Team C
2	5	2021	Champions Cup Wi…	Spain	20	Team B	Team D, Team E, Team F
3	6	1999	Won League 1999	Uganda	16	Team A	Team A, Team B, Team C, Te…
4	7	2020	Won League 2020	England	16	Team A	Team A, Team E, Team G, Te…

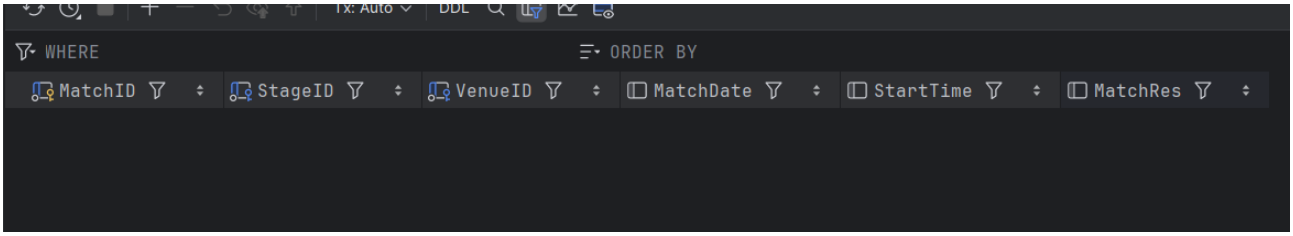


StageID	CupID	StageName	StartDate	EndDate	NumberMatches	NumberOfTeams

It’s a Stage.
Attention! It was only supposed to remove elements with StageID = 1, but 2 were removed in previous tests. So all work correctly.

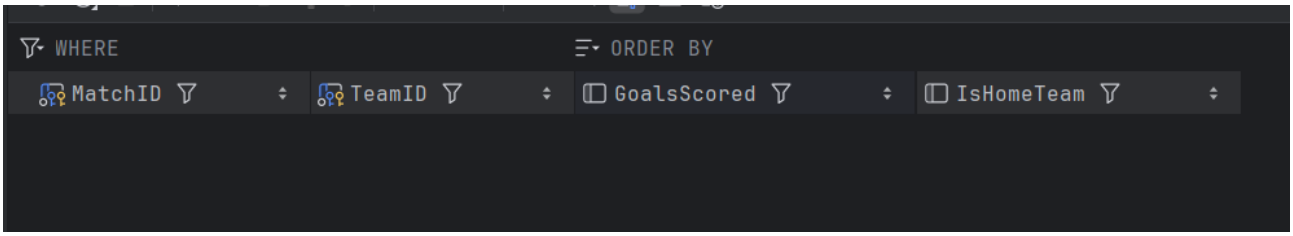
Next tables with removed data:

Match_:



MatchID	StageID	VenueID	MatchDate	StartTime	MatchRes

Matchteam:



Conclusion:

At the beginnig the usage of MySQL with DataGrip and MySQL workbench was complex for me, because I have never worked with databases before, but after some hours of work, I understood a way how to solve problems.

So using MySQL with DataGripto define and manipulate data in this domain was generally straightforward and efficient. The ability to define tables with clear relationships using foreign keys made it easy to model the domain, such as linking matches, venues, and stages. The flexibility of data types and constraints allowed me to enforce integrity rules, ensuring that data remained consistent.

When it came to data manipulation, MySQL's simple syntax for INSERT, SELECT, UPDATE, and DELETE queries made it easy to perform the necessary operations. Joins were particularly helpful for retrieving data across related tables. While performing complex DELETE operations involving foreign keys required additional consideration (like cascading actions), the overall process was manageable.

Overall, MySQL provided an expressive, reliable, and comfortable experience for handling the domain's data, though some advanced scenarios, like managing foreign key constraints during deletions, required extra attention.

I understand why we have made a conceptual model of our database. It was crucial for Homework 2