Team members:

TEAM MEMEBERS	STUDENT NUMBER	Signature
SURAJ PANDEY	NPI000051	
SANDESH SUBEDI	NPI000040	
NABIN CHHETRI	NPI000032	
SANDESH GIRI	NPI000041	

1. Table of Contents

Abstract

The main purpose of this project is to design complete database system for the online mobile game called *BingBing* legend. The project starts with the identification of business rules of the game. After the identification of business rules, entities, attributes and relationship between the entities, ERD (Entity relationship diagram) is formed. Then the ERD is converted into its corresponding relational schema and normalization is being carried out to design a perfect database system. Finally, all the tables are created using MySQL and database schema is implemented.

Acknowledgement

We would like to exhibit our genuine gratefulness to everyone who assisted us to complete this database design task. Firstly, we'd love to display extraordinary thankfulness to our subject teacher, Dr. Bidur Devkota. His counsel and encouragement lead us to complete our assignment task within proper time along with appropriate guidelines. Additionally, we'd love to applaud our mates who helped us in collecting required data. Without their cooperation, this paper would not be as good.

Disadvantages of file based system

- **Data Redundancy and Instability**:

Though data are occupying in different data files, so there is possibility of redundancy and instability in data. For example, if the same customer has 2 different saving account as well as mortgage loan. So, here the customer details can be duplicated after all the programs for two different functions keep their similar data in two different data files. So, this may result to redundancy in customer data. If the same data is kept in 2 files, then it gives rise to instability.

- **Low Security**:

Security of data is major concern in database management system. Customers or users should be only accessible to required records rather than whole database. But this is not in practice in case of file based system, due to which data may get lost. Hence, data are not secured in file based system.

- **Unexpected Queries**:

Handling sudden/ad-hoc queries is so tough in file based system. It requires change in existing programs or application. This system is relatively uncomplicated but it cannot handle complicated queries and data recovery.

- **Chances of error and time consuming**:

In file based systems, there is likely to produce errors due to modifications. Thus, it requires more and extra time to debug the programs and correct those errors.

- **Integrity Problems**:

There can be definite data integrity rules which need to be maintained in any applications or programs. It may be noted that we are not claiming to say that we cannot handles all those above problems in file management system but the real issue is that each applications or programs had to handles all those problems of their own.

Advantages of database and DBMS

1. **Improved data sharing:**

An advantage of the database management approach is that it helps to establish an environment in which more and better-managed information is allowed to user. It gives users the capacity to create more information from given data and also permit for data sharing among employees and also to other people who have access to the system.

2. **Improved data security:**

A DBMS produces a framework for better and greater fulfillment of data privacy and security policies. Each user can have a distinct set of access so data is secured from different negative issues like identity theft, leaks and misuse of data.

3. **Control of redundancy of data:**

Redundancy of data is removed by data normalization. If there is no duplication of data, then it may improve access time and also saves storage. Redundancy is controlled by application programming and kept to least by initiating as little redundancy as possible when making the database.

4. **Minimized Data Inconsistency:**

As we know, the root cause of data inconsistency is redundancy of data because of involvement in normalization of data. It exists when different version of identical data emerges in different place.

5. **Improved data access:**

Database management system makes possible to provide quick answer to ad-hoc queries. A query is a certain request provided to DBMS for data manipulation from a perspective of database. Database controls data in such a way that data can be easily

6. **Improved decision making**:

Better data management and improved data access allow quality information on the basis of better decisions. The quality of generated information associates to the quality of underlying data. Decision making results in a framework to ease data quality initiatives.

Relation with case study

Overall, Database and database management systems are more effective to run, implement and design as per the requirement for "BingBing legend" mobile game than file-based systems. It permits the user to check the data by creating, controlling, defining, insertion, deletion and revising. It makes it possible for multiple authorized users to gain the same database in different ways from different places to complete different intents.

Business rules and Normalization

Business Rules:

Business rule helps us to identify the entities, attributes and relation. This further helps us to design entity relationship diagrams. So, the business rules according to the given scenario are listed below:

- A distinctive email address and username is required during player registration.
- At least one hero must be owned prior to any skin purchase.
- Only 5 (no more or less) active-players are permitted in a team.
- Heroes must be selected uniquely within a team during the battle.

- Skins and heroes can only be procured with diamonds or battle points.

Normalization

As per the requirement of the question let us take an example of UNF (un normal form) from given scenario which is given below:

Player Name	Address	Clubs	Rank Name
Lionel Messi	Catalonia	FC Barcelona	Commander BE
Neymar Jr	Paris	Paris SG, FC Barcelona	Member BE
Alex Morgan	Madrid	Bayern CF, Real CF	Member BE
Neymar Jr	Manchester	Manchester Utd	Order BE

Table1: Un Normal Form (UNF)

The table above is an example of un normal form. It consists of multivalued attributes in same tuple due to which it lacks efficiency and may face multiple issues like data redundancy or anomalies. Hence, it should be further normalized.

1NF (First Normal Form)

To perform first normal form, we need to follow certain conditions which are listed below:

- Domains of all its attributes must be atomic.
- The value of any attribute in a record is a single valued from its domain.

So, the 1NF of given UNF example after fulfilling all the conditions of 1NF table appears as bellow:

Player Name	**Address**	**Clubs**	**Rank Name**
Lionel Messi	Catalonia	FC Barcelona	Commander BE
Neymar Jr	Paris	Paris SG	Member BE
Neymar Jr	Paris	FC Barcelona	Member BE
Alex Morgan	Madrid	Bayern CF	Member BE
Alex Morgan	Madrid	Real CF	Member BE
Neymar Jr	Manchester	Manchester Utd	Order BE

Table 2: First Normalized Form (1NF)

After the table is normalized into first normal form by following required conditions, each tuple must be distinct. In table above, there are two players with same name, therefore we need to consider both name and address to identify a tuple distinctly. Doing that will create a composite key, which is a primary key composed of numerous columns to determine a tuple distinctively.

2NF (Second Normal Form)

Conditions of 2NF are:

- Table must be in 1NF.
- Relation must not contain any partial dependency.
- Table should only contain data of one type of thing.

Partial dependency: If non-key attributes are functionally dependent on part of a candidate key then it is called partial dependency.

2NF

Player Registration No.	Player Name	Address	Rank Name
FIFA1001	Lionel Messi	Catalonia	Commander BE
FIFA1002	Neymar Jr	Paris	Member BE
FIFA1003	Alex Morgan	Madrid	Member BE
FIFA1004	Neymar Jr	Manchester	Order BE

Table 3.1 : Player Information Table

Player Registration No.	Clubs
FIFA1001	FC Barcelona
FIFA1002	Paris SG
FIFA1002	FC Barcelona
FIFA1003	Bayern CF
FIFA1003	Real CF
FIFA1004	Manchester Utd

Table 3.2 : Club Information Table

As there was no other ways to simplify the table, we need to divide the table into two different tables. With this, The first table displays information about players while the second one displays club information. A new column named ‘Player registration No.’ is introduced as a primary key so that the data can be uniquely identified.

*** Player Registration No. Is the foreign Key in Club Information table**

In Club Information Table (Table 3.2), Player Registration Table is the foreign key.

A foreign key is basically a set of attributes that is accustomed as a reference of primary key in discrete table. It is used to connect two tables together. In Table 3.2, Player Registration No. Is referred as a foreign key which was primary key column in Table 3.1 (Player Information Table).

3NF (Third Normal Form)

Condition 3NF are:

- Table must be in 2NF
- Must not contain transitive dependency.
-

Transitive dependency: If any non-key attribute is functionally dependent up on another non key attribute then the relation is called transitive dependency.

Player Registration No.	**Player Name**	**Address**	**Rank Name**
FIFA1001	Lionel Messi	Catalonia	Commander BE
FIFA1002	Neymar Jr	Paris	Member BE
FIFA1003	Alex Morgan	Madrid	Member BE
FIFA1004	Neymar Jr	Manchester	Order BE

Table 4.1

There is transitive dependency between Player's Name and Rank Name. The Rank Name is dependent upon Player Name. Modifying the name of player may alter the rank name of the player too. Hence, there is transitive dependency present in the table which does not fulfill the condition of Third Normal Form.

Conversion into 3NF,

Player Registration No.	Player Name	Address	Rank Name ID
FIFA1001	Lionel Messi	Catalonia	CBE-1
FIFA1002	Neymar Jr	Paris	MBE-1
FIFA1003	Alex Morgan	Madrid	MBE-1
FIFA1004	Neymar Jr	Manchester	OBE-1

Table 4.2

Player Registration No.	Clubs
FIFA1001	FC Barcelona
FIFA1002	Paris SG
FIFA1002	FC Barcelona
FIFA1003	Bayern CF
FIFA1003	Real CF
FIFA1004	Manchester Utd

Table 4.3

Rank Name ID	Rank Name
CBE-1	FC Barcelona
MBE-1	Paris SG
MBE-1	FC Barcelona
OBE-1	Bayern CF

Table 4.4

Following rules and conditions of Third Normal Form, the table in 2NF is divided and new table is created. The new table stores Rank Name of each player. The Rank Name ID in Table 4.3 while it is foreign key in Table 4.1.

In this way a table is normalized up to Third Normal Form in order to avoid redundancies and data anomalies. There is no transitive dependency I.e, the table is in Third Normal Form.

Entity Relationship Diagram(ERD)

"Entity Relationship diagram is a pictorial representation that outlines relationships among people, objects, places, concepts or events within an information technology(IT) system" (B. Jacqueline,2019). Before drawing entity relationship diagrams for database management we need to go through certain steps which helps us in developing perfect ERD. The steps are as follows:

Step 1: Identify entities.
Step 2: Identify attributes of entities.
Step 3: Identify relationships between the entities.

If we are able to perform all the steps mentioned above, then it will be easier for us to design ERD of the system.

Identification of entities

Entities:(definition)

First of all, we need to identify the entities present in a given case study. Entities are represented by rectangular shape in entity relationship diagram. The entities available in the given scenario are as follows:

ENTITIES

a. Player

b. account

c. Hero

d. Skin

e. Game-mode

f. Team

g. Personal statistics

h. Game

Description of each entity:

Entities	description
player	Information of any user of this game will get through a unique player account.
Hero	Information about heroes can be owned by players and can be used as per role that user chooses in the game.
Skin	The player of this game can equip details of skin for each hero.
Game_mode	All the game-mode is stored by an entity that can be picked and played by the team.
Game	This entity holds the record of result and number of participants that played in this game.

Team	This entity is related to creating a team consisting five players and the heroes which they picked with their respective levels. In this case, this entity creates game mode very simply and efficiently.
Personal statistics	This entity holds the details of player's and accomplishments.

Identification attributes of entities

After identifying the entities present in the given case study we need to find out the attributes of those entity sets which describe the characteristics of each entity present in the entity sets. So the attributes of the above listed entity sets are as follows:

ATTRIBUTES:

1. **Player:**

    a) Player_ID
    b) Player_name
    c) Email
    d) Username
    e) Global_level
    f) Diamonds collected
    g) Online_status
    h) Hero_lvl
    i) skin_owned

2. **hero**

    a) Hero_ID
    b) Hero name
    c) Hero_Role
    d) Hero_Speciality
    e) Hero_price

3. **Skin**
    a) Skin_ID
    b) Skin name
    c) Skin price

4. **Game_mode**
    a) Game_mode ID
    b) Game_mode type
5. **Game**
    a) Game ID
    b) Game_date
    c) battle_duration
    d) Game result
    e) Arena

6. **Team**
    a) Team_ID

7. **Personal statistics**
    a) Result history
    b) Game won
    c) Additional detail
    d) nationality

Description of attributes of each entity

Entity : player

Attributes	description
Player_ID	A distinctive ID given to each player which distinctly identify each information stored on that table of that particular player.
Player_name	To make the message dynamic, the player's real name is used while sending the player emails and updates.
Email	Email address of player.
Username	Player’s name in game.
Global_level	The global level of player’s viewed publicly.
Diamond Collected	This holds the record of collection of diamonds by players.
Online_Status	It is the boolean process where true indicates active or online and false indicates inactive or offline.

Entity : Hero

Attributes	description
Hero_ID	A unique primary key indicates each hero and automatically increments.
Hero_name	The names of hero’s like Fanny and Estes, Clint, Cyclops, Rafaela, Dark Rose, etc.
Hero _Role	Role description of hero’s like Assassin, Fighter, etc.
Hero_Speciality	Specialty description of hero’s like Regen, Charge, etc.
Hero_Price	It is the integer which represents the price of hero’s in diamonds and battle points.
Hero_lvl	Specific played owned their current level of a specific hero.

Entity : Skin

Attributes	description
Skin_ID	A unique primary key which indicates each skin and automatically increments.
Skin_name	A name of skin is usually one to describe it.
Skin_price	It is the integer which represents the price of skin's in diamonds and battle points.

Entity : Game_mode

Attributes	description
Game_mode_ID	A unique primary key which indicates each available game mode and automatically increments.
Game_mode type	Classic, Brawl, or human vs AL mode are categorized as specific types of the mode.

Entity : Game

Attributes	description
Game_ID	A unique primary key which indicates to each that has been played by players and automatically increments.
Game_date	It indicates the date variable in which the game shows the date that has been started.
battle _duration	It is an integer which represents the duration of game in which game can be change it into appropriate form if needed
Game_result	A foreign key which represents the team_ ID of the team that lost or won the game.
Arena	It is the cosmetics and environment of each game mode.

Entity : Team

Attributes	**description**
Team_ID	A unique primary key which indicates each team created and automatically increments.

Entity : Personal Statistics

Attributes	**Description**
Player_ID	It is a foreign key which indicates the Player_ID that is located in the player in which the player has been linked to their statistics.
Most_used_hero	It is a foreign key which indicates the Hero_ID that is located in the hero entity in which the player used the hero most.
pervious_game_result	It is represented by using boolean in which false indicates the player lost the game and true indicates the player one last game they played.
Additional_result	Player description with their profiles.

Simple ERD (Entity Relationship Diagram)

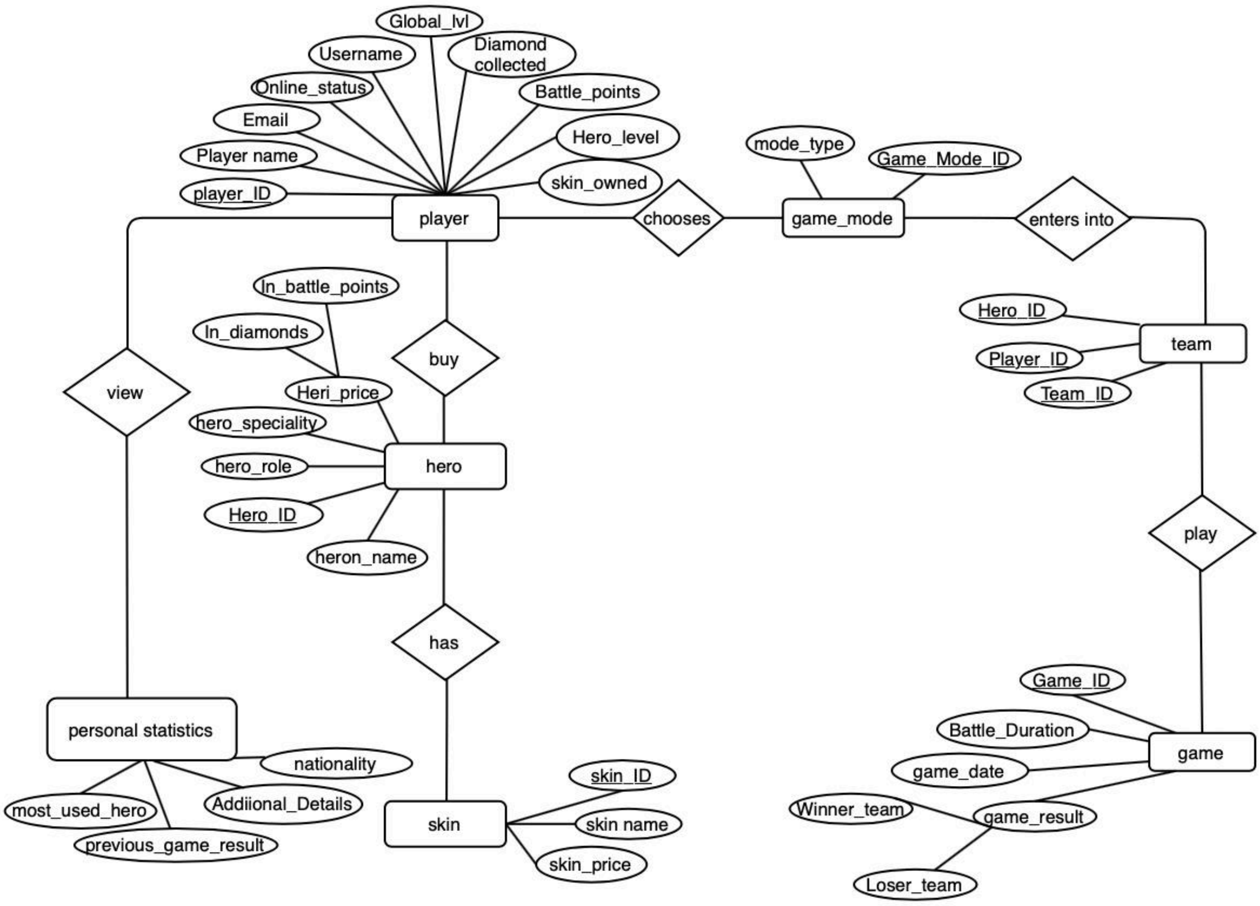


Figure: Simple ERD without cardinality

Identification of relationship between entity

To draw an entity relationship diagram we need to find out the relation between two different entities which helps us in building ER diagrams easily. There are mainly four types of relation that associate entities with each other which are describe below:

One to one (1:1)

If one entity of the first entity set is associated with only one entity of another entity set then the relation is called one to one relation.

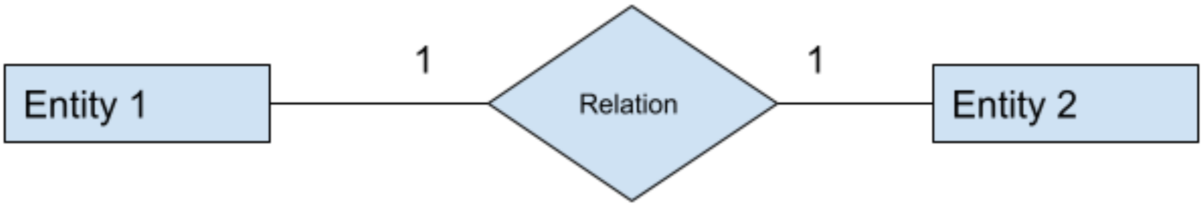


Fig: one to one relation

One to many (1:M)

If one entity of the first entity set is associated with more than one entity of another entity set is called one to many relation.

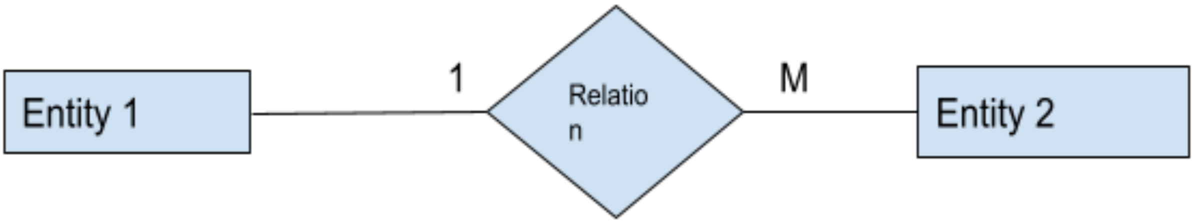


Fig: one to many relation

Many to one (M:1)

Similarly, it is a type of relation in which many entities of the first entity set are linked with only one entity of another entity set .

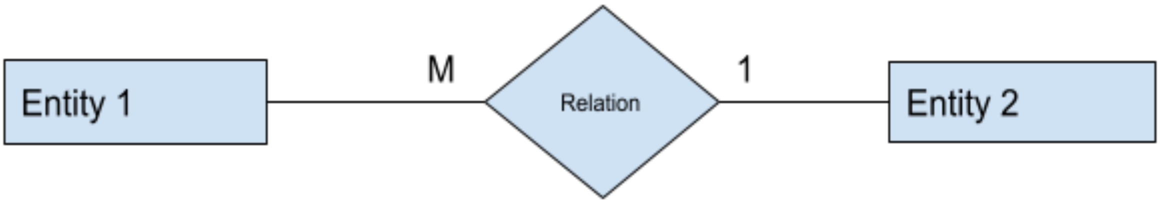


Fig: many to one relation

Many to many(M:M)

If many entities of the first entity are linked with more than one entity of another entity set, then the relation is said to be in many to many relations.

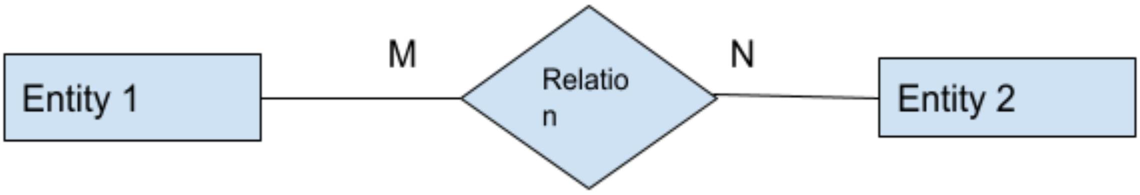


Fig: many to many relation

The relationship between entity according to the given scenario are listed below:

relationship	description
Player M:M Hero	One player can buy many heroes; one hero can be bought by many players. There exists many to many relationships between player and hero.
Hero 1:M skin	One hero can have many skins but a skin is associated with only one hero.
Player 1:M Personal statistics	A player has more than one personal statistics of different games that he has played.
Team M:1 Game_mode	Many teams can choose the same Game_mode but only one Game_mode can be selected by a team for a game.
Team M:1 game	A game can be played by only two teams and two teams are compulsory to play a game.
Player M:N Game_mode	A player can choose many game modes and a Game_mode can also be chosen by many players.
Game_mode 1:M Game	One game is played in only one game mode and the same Game_mode can be chosen to play different games.

Constraints

Logical constraints

a. The player can buy the skin for a specific hero after getting at least one hero.

b. Using diamonds or points obtained from battle, both skin and heroes can be purchased.

c. Before the game has started, game mode must be chosen.

d. Member requirement is only five in a team.

e. Active players can get a chance in a team.

f. Each team requires a unique hero.

Null constraints

All attributes hold data which is important for an entity and should be ready to facilitate when required, so they cannot be null. Following expressions shows the importance attributes:

a. Personal_statistics table has the details of attributes that can be Null at the beginning of the game when the player will not retain details.

b. Personal_statistics table has the Most_used_hero can be Null at the starting of the game when the player will not have used any heroes.

Unique Constraints

Following are list of unique constraint present in given case study:

a. Player's name

b. Username

c. Email

d. Hero's

e. Skin

f. Both the game mode name and Arena

g. Requirement members in team.

Primary Key Constraints

a. All the primary keys are integers.

b. To be an integer, the primary key can increment automatically with start at 1 and iterate by 1, which makes insertion process easier.

Default Constraints

a. The default value of the global level attributes of the player entity is 1.

b. The default value of the Diamond attribute of the player entity is 250.

c. The defau;t value of the Battle_points attribute of the player entity is 3000.

d. The default value of [v] attribute of the player entity is 1.

Workload matrix

ASSIGNMENT COMPONENT	SURAJ PANDEY	SANDESH SUBEDI	NABIN CHHETRI	SANDESH GIRI
Advantages and disadvantages	25%	25%	25%	25%
Business rules	25%	25%	25%	25%
Normalisation	25%	25%	25%	25%
Entity Relationship Diagram	25%	25%	25%	25%
Signature	Suraj	Sandesh	Nabin	Sandesh

Database Schema

1. Finalized ERD (Entity Relationship Diagram)

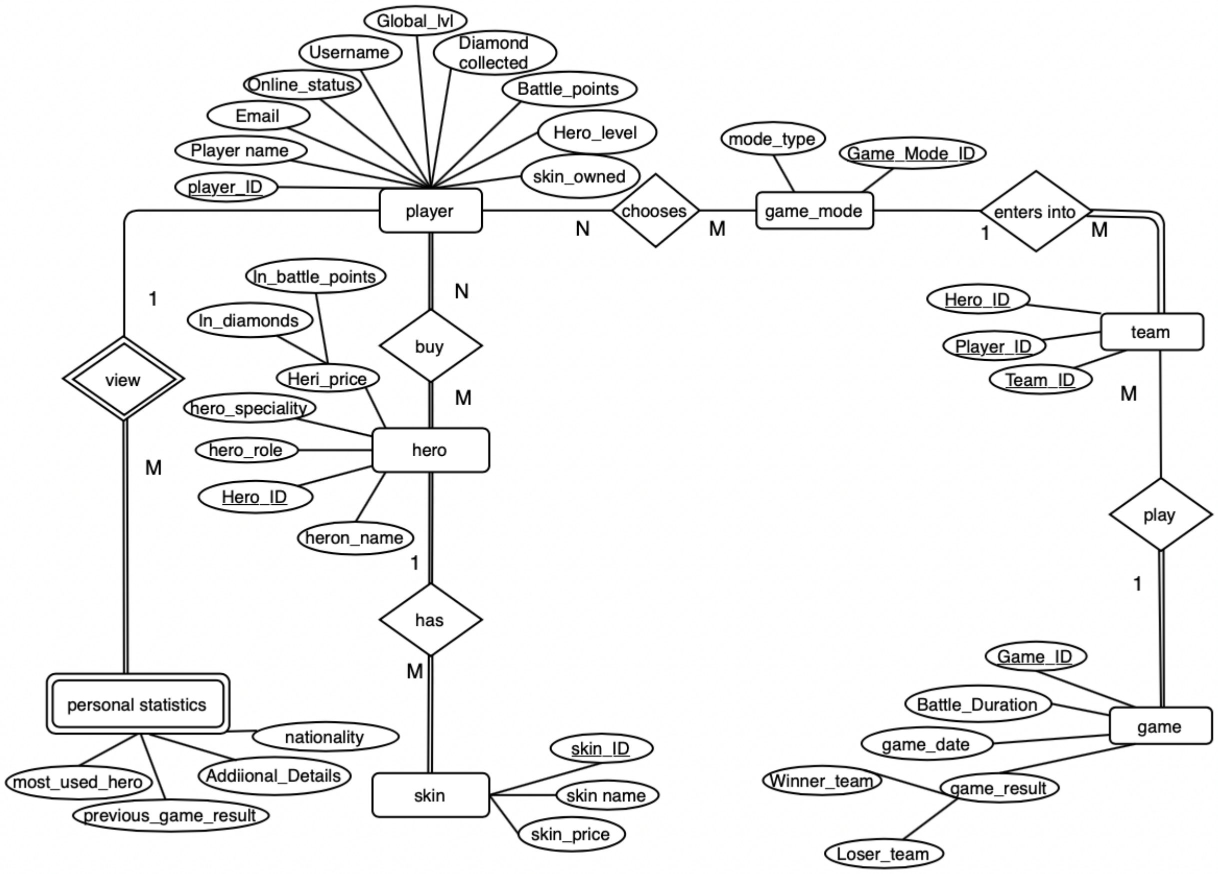


Figure: Finalized ERD (Entity Relationship Diagram)

2. Conversion of ERD (Entity Relationship Diagram) into Relational Schema

There are certain steps for the conversion of ERD (Entity relationship diagram) to relational schema. While converting ERD to its corresponding relational schema there may have formed more tables than the entity presents in the diagram. This statement gets cleared after reading the following steps. The steps are:

1. First of all, we need to convert each strong entity to its corresponding relation or table.

2. Secondly,convert attributes of the entity as columns for the table.

3. Convert multivalued attribute into new table and add primary key of the parent table as foreign key within the table.

4. Convert weak entities into tables and add each attribute to the table as column and primary key of the identifying entity.

5. Declare foreign key constraints for the table according to the relationship that exists between the entities.

* If a 1:1(one to one) relationship exists between entities then add the primary key entity of one table as foreign key to another table or vice versa.

* If total and partial participation exist between two entities in 1:1(one to one) relationship then put the primary key of the entity of partial participation as foreign key to the table with total participation.

* If 1:M (one to many) or M:1(many to one)relationships exist between entities then add the primary key of the table with one cardinality as foreign key to the table with many cardinalities.

* If M: N (many to many) relation exists between the entities, then create an intermediary table and add the primary key of both the tables as columns for the intermediary table. The combination of both keys makes the primary key which distinctly recognize each tuple of the table.

In this way, the ERD is converted into a relational schema. So, from the above steps it is clear that there may appear more tables than the entity presents in the ERD (Entity Relationship Diagram).

Conversion of strong Entity into Relation schema from ERD shown above:

Entity is converted into table and attributes as fields of the table.

- **Player entity into Player table.**

  Player table:

Player_ID	**Player name**	**Email**	**Online_status**	**Global_level**	**Diamonds_collected**	**Username**	**Hero_level**	**Skin_owned**

Primary key: Player_ID

- **Hero entity into Hero table.**

  Hero table:

Hero_ID	**Hero_name**	**Hero_role**	**Hero_speciality**	**In_diamonds**	**In_battle_points**

Primary key: Hero_ID

Similarly, Hero_price attribute has composite attributes that is price In_diamonds and price In_battle_points, so the composite attributes are illustrated into the relational schema instead of Hero_price attribute according to the rules.

- **Skin entity into Skin table.**

  Skin table:

Skin_ID	**Skin_name**	**In_diamonds**	**In_battle_points**	**Hero_ID***

Primary key: Skin_ID

Foreign key: Hero_ID

As there is one to many relations between hero table and skin table we need to add the primary key attribute of hero table as foreign key attribute to skin table which inter connect two tables with each other. Similarly, Skin_price attribute has composite attributes that is price In_diamonds and price In_battle_points, so the composite attributes are illustrated into the relational schema instead of Skin_price attribute according to the rules.

- **Game_mode entity into Game_mode table**

Game_mode table:

Game_mode ID	**Mode_type**

Primary key: Game_mode_ID

- **Team entity into Team table.**

Team table:

Team_ID	**Game_mode ID***	**Game_ID***

Primary key: Team_ID

Foreign key: Game_mode_ID, Game_ID

As shown in above figure there is one to many relations between Game_mode entity and team entity respectively. So, according to rule, we added a column called Game_mode ID which is also a primary key of Game_mode table to team table as foreign key. Similarly, there is one to many relations between game entity and team entity so Game_ID also became foreign key for team table.

- **Game entity into Game table**

Game table:

Game_ID	Battle_Duration	Winner	Loser	Game_date	Game_mode_ID*

Primary key: Game_ID

Foreign key: Game_mode_ID

➢ **Table formed due to many to many relations between entities:**

- **Player hero table formed due to many to many relations between player and hero entity.**

Player_hero table:

PH_ID	Player_ID*	Hero_ID*

Primary key: PH_ID (combination of Player_ID and Hero_ID)

Foreign key: Player_ID, Hero_ID

- **Mode_chosen table is formed due to many to many relations between Player and Game mode entity.**

Mode_chosen table**:**

Player_ID*	Game_mode_ID*

Primary key: combination of both Player_ID and Game_mode_ID

Foreign key: Player_ID, Game_mode_ID

- **Conversion of weak entity into its corresponding relation:**

In ERD (Entity Relationship Diagram) there is a weak entity called Personal_statistics table which doesn't contain its primary key. So, it borrows that primary key of player table as its own primary key which uniquely identify each row of the table.

- Personal_statistics table:

Player_ID*	**most_used_hero**	**previous_game_result**	**Additional_details**	**Nationality**

Primary key: Player_ID

Foreign Key: Player_ID

Normalization of all relation up to 3NF:

1. UNF player relation:

Player_ID	**Player_name**	**Email**	**Username**	**Online_status**	**Global_level**	**Diamonds_collected**	**Battle_points**	**Hero_ID**	**Hero_level**	**Skin_owned**
1	Suraj	suraj1@gmail.com	Suraj123	1	12	223	2223	H001, H002	12,23	Sk001, Sk002

1NF:

So the 1NF of above UNF player relation is :

Player_ID	**Player_name**	**Email**	**Username**	**Online_status**	**Global_level**	**Diamonds_collected**	**Battle_points**	**Hero_ID**	**Hero_level**	**Skin_owned**
1	Suraj	suraj1@gmail.com	Suraj123	1	12	223	2223	H001	12	Sk001
1	Suraj	suraj1@gmail.com	Suraj123	1	12	223	2223	H002	23	Sk002

2NF:

Partial dependency:

Player_ID-> Player_name, Email, Username, Online_status, Global_level, Diamonds_collected, Battle_points

Player_ID, Hero_ID ->Hero_level

Player_ID, Hero_ID->Skin_ID

Table formed after 2NF are:

Player_table:

Player_ID	**Player_name**	**Email**	**Username**	**Online_status**	**Global_level**	**Diamonds_collected**	**Battle_points**
1	Suraj	suraj1@gmail.com	Suraj123	1	12	223	2223

Player_hero table:

Player_ID*	**Hero_ID***	**Hero_level**
1	H001	12
1	H002	23

Player_skin table:

Player_ID*	**Hero_ID**	**Skin_ID***
1	**H001**	Sk001
1	**H002**	Sk002

Since, there is not any transitive dependency present in the player table. Hence,3NF player table is :

Player_ID	**Player_name**	**Email**	**Username**	**Online_status**	**Diamonds_collected**	**Battle_points**
1	Suraj	suraj1@gmail.com	Suraj123	1	223	2223

2. Hero table:

Hero_ID	**Hero_name**	**Hero_role**	**Hero_speciality**	**In_diamonds**	**In_battle_points**
H001	Rafaela	Fighter	Charge	240	1500
H002	Fanny	Assassin	Reap	350	2000

Hero_ID ->Hero_name, Hero_role, Hero_speciality, In_diamonds, In_battle_points. {full dependency}.

Hence, there is not any partial and transitive dependency present. So, the table is already in 3NF.

3.Skin table:

Skin_ID	**Skin_name**	**Skin_price**	**Hero_ID***
Sk001	Recon	200	H001
Sk002	Glacier	250	H002

Skin_ID -> Skin_name, Skin_price Hero_ID*. {full dependency}.

Hence, there is not any partial as well as transitive dependency. So, the table is already in 3NF.

4. Relation: Game_mode

Game_mode_ID	**Mode_type**
GM001	Brawl
GM002	Human vs AI

Game_mode_ID -> Mode_type {full dependency}

Hence, there is not any partial as well as transitive dependency. So, the table is already in 3NF.

5. Relation: Team

Team_ID	**Player_ID***	**Game_ID***	**Game_mode_ID***
1	**2,4,7,3,5**	1	GM002

Since, there are repeating data in player_ID column. Above table is in UNF (Un Normal Form).

1NF:

Team_ID	**Player_ID***	**Hero_ID***	**Game_ID***	**Game_mode_ID***
1	2	H002	1	GM002
1	3	H004	1	GM002
1	4	H003	1	GM002
1	5	H008	1	GM002
1	7	H001	1	GM002

Candidate key: Team_ID, Player_ID

Partial dependency:

Team_ID -> Game_ID, Game_mode_ID

Player_ID ->Hero_ID

2NF:

To make above table into 2NF, table need to be spilt into two table as below:

Team table

Team_ID	**Game_ID***	**Game_mode_ID***
1	1	GM002

Team_player table:

Team_ID	**Player_ID***	**Hero_ID***
1	2	H002
1	3	H004
1	4	H003
1	5	H008
1	7	H001

Hence, there is no transitive dependency present in above tables the tables are already in 3NF .

6. Relation: Game

Game_ID	**Battle_duration**	**Winner_team**	**Loser_team**	**Game_date**	**Arena**	**Game_mode_ID***
1	00:30:12	1	2	2020-02-01	TDM	GM002

Game_ID -> Battle_duration, Winner_team, Loser_team, Game_date, Game_mode_ID {**full dependency**}

Since, there is no any partial as well as transitive dependency presence in above relation, the relation is already in 3NF.

7. Relation: Player_hero

PH_ID	**Player_ID***	**Hero_ID***
PH001	1	H002
PH002	1	H003

Since, there are no any non-key attributes present in the table, table is already in 3NF.

8. Mode_chosen

Player_ID*	**Game_mode_ID***
1	GM002
2	GM003
1	GM003

since there are no any non key attributes present in the table . Hence, the table is already in 3NF(Third Normal Form).

9. Relation: Personal_statistics

Player_ID*	**most_used_hero**	**Previous_game_result**	**Additional_result**
1	H001	Win	score=25
2	H002	Lose	score=0

Since, there is no any partial as well transitive dependency in above table , Personal_statistics table is already in 3NF.

Table formed after normalisation up to 3NF are:

1. Player table:

Player_ID	**Player_name**	**Email**	**Username**	**Online_status**	**Global_level**	**Diamonds_collected**	**Battle_points**
1	Suraj	suraj1@gmail.com	Suraj123	1	12	223	2223

2. Hero table:

Hero_ID	**Hero_name**	**Hero_rolle**	**Hero_speciality**	**In_diamonds**	**In_battle_points**
H001	Rafaela	Fighter	Charge	240	1500
H002	Fanny	Assassin	Reap	350	2000

3. Skin table:

Skin_ID	**Skin_name**	**Skin_price**	**Hero_ID***
Sk001	Recon	200	H001
Sk002	Glaciar	250	H002

4. Game_mode table:

Game_mode_ID	**Mode_type**
GM001	Brawl
GM002	Human vs AI

5. Team table:

Team_ID	**Game_ID***	**Game_mode_ID***
1	1	GM002

6. Game table:

Game_ID	**Battle_duration**	**Winner_team**	**Loser_team**	**Game_date**	**Arena**	**Game_mode_ID***
1	00:04:12	1	2	2020-02-01	TDM	GM002

7. Team_player table:

Team_ID	**Player_ID***	**Hero_ID***
1	2	H002
1	3	H004
1	4	H003
1	5	H008
1	7	H001

8. mode_chosen table

Player_ID*	**Game_mode_ID***
1	GM002
2	GM003
1	GM003

9. Player_hero table

PH_ID	**Player_ID***	**Hero_ID***	**Hero_level**
PH001	1	H001	12
PH002	1	H002	23

10. hero_skin table:

Player_ID*	**Hero_ID**	**Skin_ID***
1	H001	Sk001
1	H002	Sk002

Since, the combination of both Player_ID and Hero_ID makes PH_ID, we can replace Player_ID and Hero_ID by PH_ID. So, the hero_skin table changes to:

PH_ID	**Skin_ID***
PH001	Sk001
PH002	Sk002

11. personal_statistics table:

Player_ID*	**most_used_hero**	**Previous_game_result**	**Additional_result**
1	H002	Win	score=25
2	H003	Lose	score=0

Data Dictionary:

Player table

Attributes	Description	datatype	Constraint	Key
Player_ID	Unique id given to each player to uniquely identify the each row of the table.	Int	Not null, Unique, Auto increment	Primary key
Player_name	Each player's name	Varchar	Not null	
Username	Player's game name	Varchar	Not null, Unique	
Email	Player's email address	Varchar	Not null, Unique	
Online_status	Stores the information regarding player's status like whether they are online or offline for example if a player is online then shows 1 else shows 0.	BIT		
Global_level	player's level in game	Int	Not null	
Diamonds_collected	Holds the information about the total number of diamonds that player has currently.	Int		

Hero table:

Attributes	Description	datatype	Constraint	Key
Hero_ID	Unique ID of each hero that uniquely identify each hero.	Varchar	Not null, unique	Primary key
Hero_name	Name given to each hero like Rafaela, fanny, Estes, etc.	Varchar	Not null, unique	
Hero_role	Role given to the hero like fighter, tank, assassin, etc.	Varchar	Not null	
Hero_speciality	Special power that hero own.	Varchar	Not null	
In_diamonds	Price of hero in diamonds.	Int		
In_battle_points	Price of hero in battle points	Int		

skin table:

Attributes	Description	datatype	Constraint	Key
Skin_ID	ID of each skin that uniquely distinguish each skin name.	Varchar	Not null, unique	Primary key
Skin name	Name given to the skin.	Varchar	Not null	
Skin_price	Price of each skin.	Int	Not null	
Hero_ID*	ID of hero that own skin.	varchar	Not null	Foreign key

Player_hero table:

Attributes	Description	datatype	Constraint	Key
PH_ID	Unique ID that identify each tuple distinctly, which is the	Varchar	Not null, unique	Primary key

Attributes	Description	datatype	Constraint	Key
	combination of Player_ID and Hero_ID.			
Player_ID*	ID of the player that owned hero.	Int	Not null	Foreign key
Hero_ID*	ID of the hero owned by the player.	Varchar	Not null	Foreign key
Hero_level	Each player's hero level.	Int	Not null	

Hero_skin table:

Attributes	Description	datatype	Constraint	Key
PH_ID*	ID that represent the specific hero of the specific player .	Varchar	Not null	Primary key, Foreign key
Skin_ID*	ID of the skin that player's hero owned.	Varchar	Not null	Primary key, Foreign key

Game_mode table:

Attributes	Description	datatype	Constraint	Key
Game_mode ID	Unique ID that identify each game_mode uniquely.	Varchar	Not null, unique	Primary key
Mode_type	Types of mode available in the game.	Varchar	Not null	

Team table:

Attributes	Description	datatype	Constraint	Key
<u>Team_ID</u>	Unique ID of each team played in each game which is automatically created.	Int	Not null, unique	Primary key
game_mode ID*	The ID of the mode in which the team has played the game.	Varchar	Not null	Foreign key
Game_ID*	The ID of the game in which team has played.	Int	Int	Foreign key

Game_table:

Attributes	Description	datatype	Constraint	Key
<u>Game_ID</u>	Unique ID formed automatically when game is played.	Int	Not null, unique	Primary key
Battle_duration	Total time that the game has run. Holds the time in seconds.	Int	Not null	
Winner_team	The game winning team.	Int	Not null	Foreign key,
Loser_team	The team that loses team.	Int	Not null	Foreign key
Game_date	The date in which the game was started.	Int	Not null	
Arena	The scene in which the game is played.	varchar	Not null	

Attributes	Description	datatype	Constraint	Key
Game_mode_ID*	The unique Id of the mode in which the game is being played.	Varchar	Not null	Foreign key

Team_player table:

Attributes	Description	datatype	Constraint	Key
Team_ID*	ID of the team .	Int	Not null	Foreign key, primary key
Player_ID*	ID of the each player participation in the team.	Int	Not null	Foreign key, primary key
Hero_ID*	ID of the hero that team's player had chosen.	Varchar	Not null	Foreign key, primary key

Mode_chosen:

Attributes	Description	Datatype	Constraint	Key
Player_ID*	ID of the player that chooses Game_mode.	Int	Not null	Foreign Key
Game_mode_ID*	ID of the game mode choose buy the player.	Varchar	Not null	Foreign Key

Personal statistics:

Attributes	**Description**	**datatype**	**Constraint**	**Key**
Player_ID*	ID of the player	Int	Not null	Primary key, foreign key
Most_used_hero	Holds the record about the hero that has been used most of the time in the game by player.	Varchar	Not null	
Previous_game_result	Result of previous games whether the game is won or lose.	Varchar	Not null	
Additional_details	About player's profile details.	Varchar	Not null	

Database Diagram:

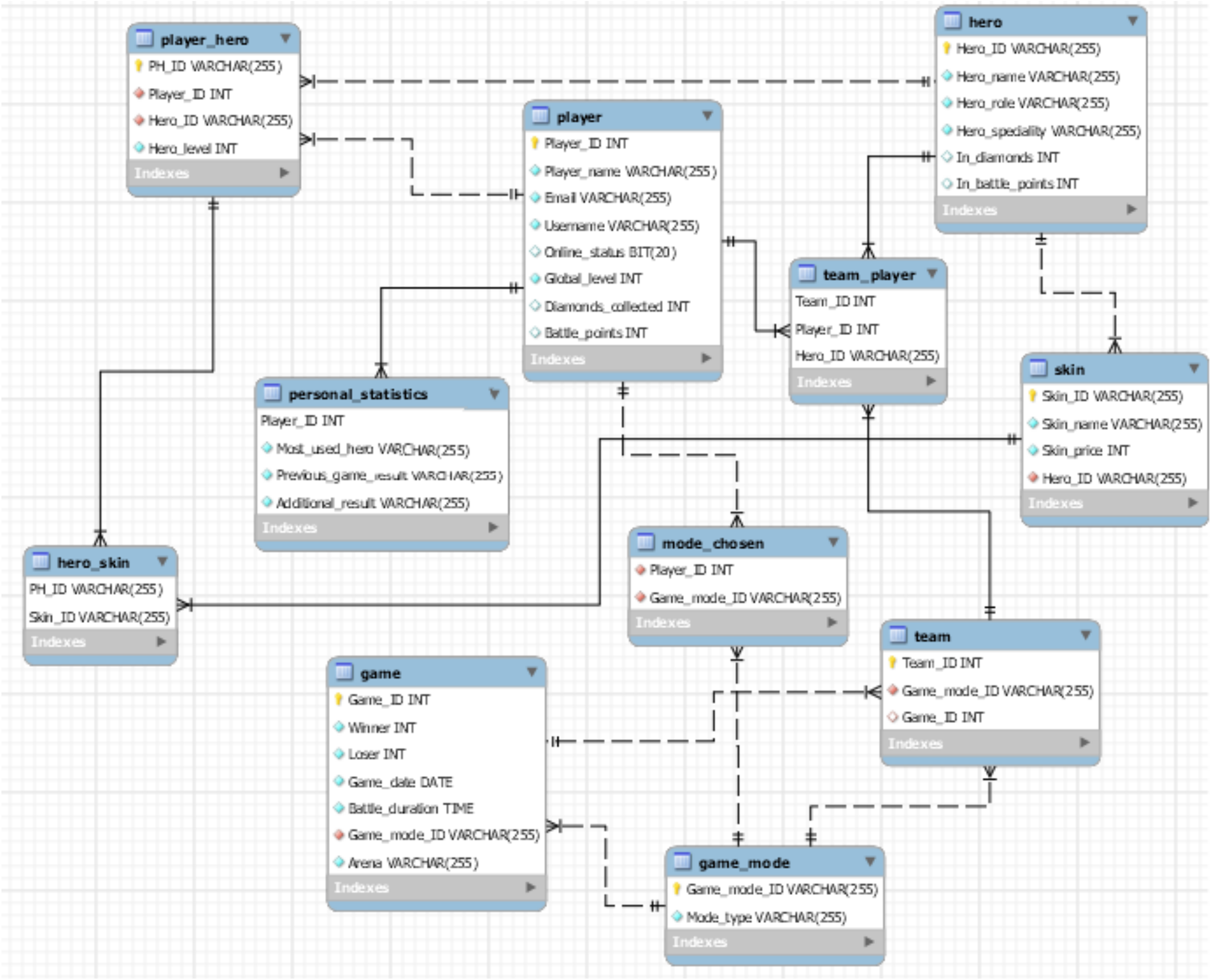


Figure: Database Diagram.

SQL-Data Definition Language SQL-DDL

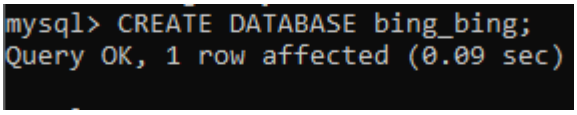
There are various syntaxes for the creating of database, creating of tables, insertion of data in the table, update data, delete data. All syntaxes are described below one by one.

Create database:

In SQL, firstly we need to create a database. Without creating database, we cannot create tables. The syntax that is used for the creating of database in MySQL server is:

CREATE DATABASE database_name;

DATABASE bing_bing:



```
mysql> CREATE DATABASE bing_bing;
Query OK, 1 row affected (0.09 sec)
```

<u>Create tables:</u>

Same like database we need to create table where we can insert data. Tables is creating inside the database which you have previously created. So, to create a table you need to first execute the ‘use <database_name in which you want to create a table>' command. Then the you need to create a table using following syntax:

CREATE TABLE table_name(

Column_1 datatypes,

Column_2 datatypes,

column_3 datatypes,

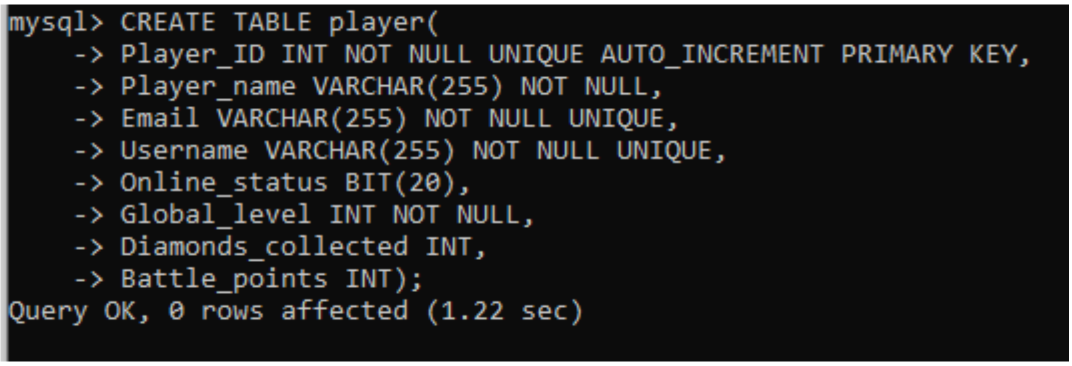
….

PRIMARY KEY(Column_name),

FOREIGN KEY(Column_name) REFERENCES table_name(Column_name)

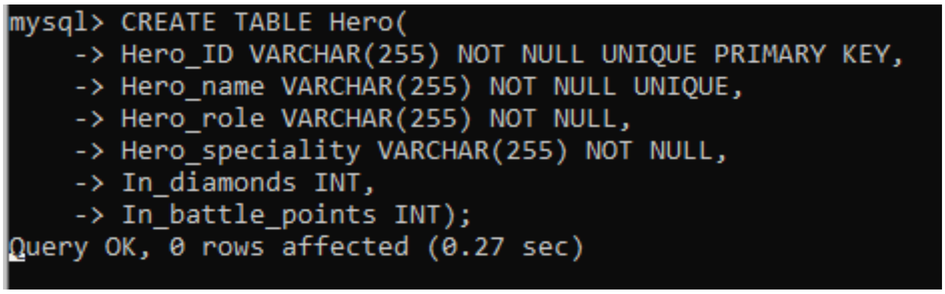
);

1. **Player_table:**



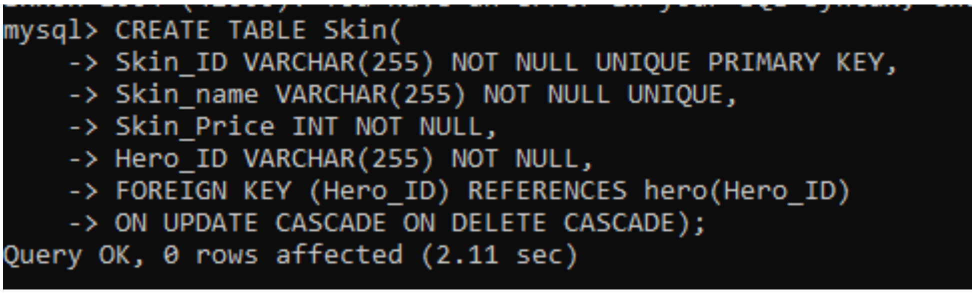
```
mysql> CREATE TABLE player(
    -> Player_ID INT NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,
    -> Player_name VARCHAR(255) NOT NULL,
    -> Email VARCHAR(255) NOT NULL UNIQUE,
    -> Username VARCHAR(255) NOT NULL UNIQUE,
    -> Online_status BIT(20),
    -> Global_level INT NOT NULL,
    -> Diamonds_collected INT,
    -> Battle_points INT);
Query OK, 0 rows affected (1.22 sec)
```

2. **Hero_table:**



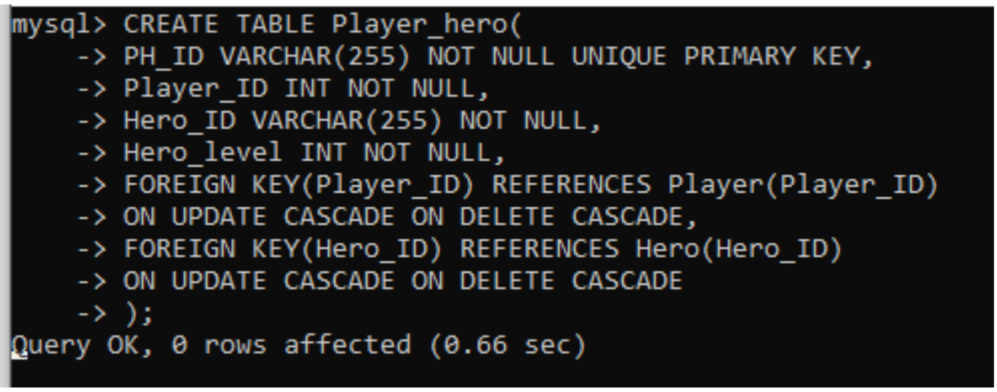
```
mysql> CREATE TABLE Hero(
    -> Hero_ID VARCHAR(255) NOT NULL UNIQUE PRIMARY KEY,
    -> Hero_name VARCHAR(255) NOT NULL UNIQUE,
    -> Hero_role VARCHAR(255) NOT NULL,
    -> Hero_speciality VARCHAR(255) NOT NULL,
    -> In_diamonds INT,
    -> In_battle_points INT);
Query OK, 0 rows affected (0.27 sec)
```

3. **Skin_table**:



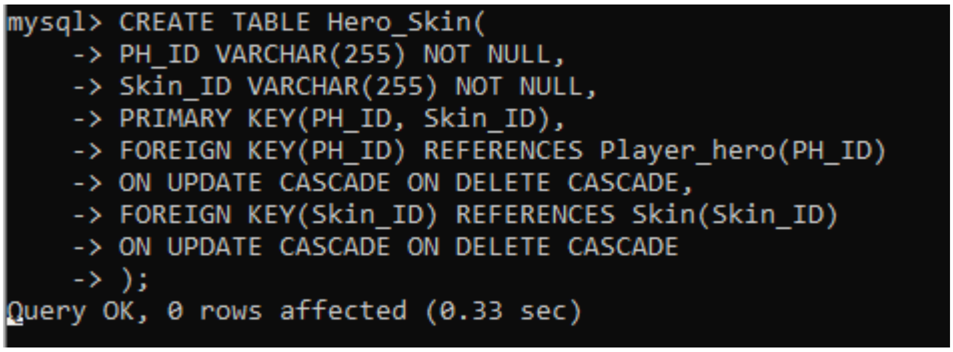
```
mysql> CREATE TABLE Skin(
    -> Skin_ID VARCHAR(255) NOT NULL UNIQUE PRIMARY KEY,
    -> Skin_name VARCHAR(255) NOT NULL UNIQUE,
    -> Skin_Price INT NOT NULL,
    -> Hero_ID VARCHAR(255) NOT NULL,
    -> FOREIGN KEY (Hero_ID) REFERENCES hero(Hero_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE);
Query OK, 0 rows affected (2.11 sec)
```

4. **Player_hero table:**



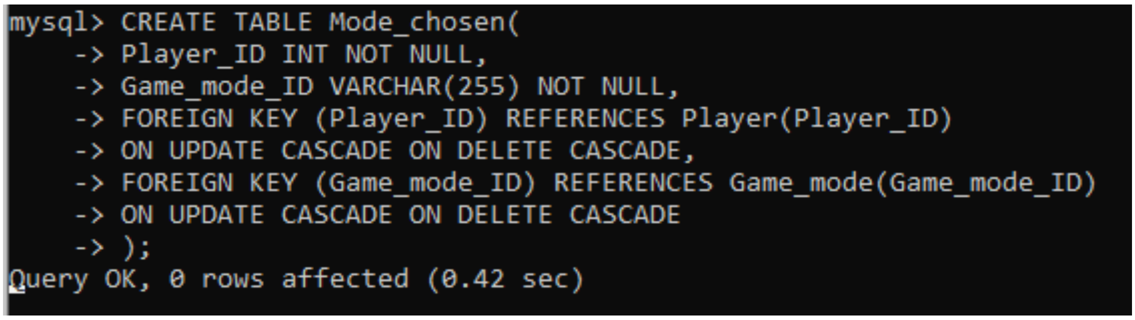
```
mysql> CREATE TABLE Player_hero(
    -> PH_ID VARCHAR(255) NOT NULL UNIQUE PRIMARY KEY,
    -> Player_ID INT NOT NULL,
    -> Hero_ID VARCHAR(255) NOT NULL,
    -> Hero_level INT NOT NULL,
    -> FOREIGN KEY(Player_ID) REFERENCES Player(Player_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE,
    -> FOREIGN KEY(Hero_ID) REFERENCES Hero(Hero_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.66 sec)
```

5. **Hero_skin table:**



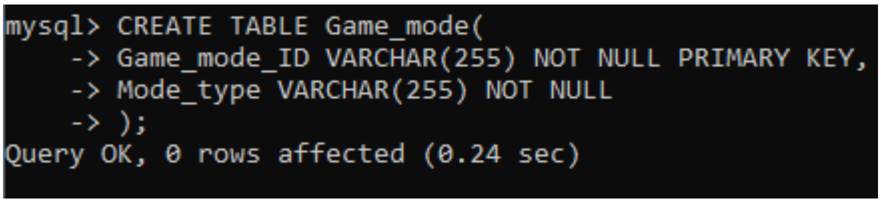
```
mysql> CREATE TABLE Hero_Skin(
    -> PH_ID VARCHAR(255) NOT NULL,
    -> Skin_ID VARCHAR(255) NOT NULL,
    -> PRIMARY KEY(PH_ID, Skin_ID),
    -> FOREIGN KEY(PH_ID) REFERENCES Player_hero(PH_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE,
    -> FOREIGN KEY(Skin_ID) REFERENCES Skin(Skin_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.33 sec)
```

6. **Mode_chosen table:**



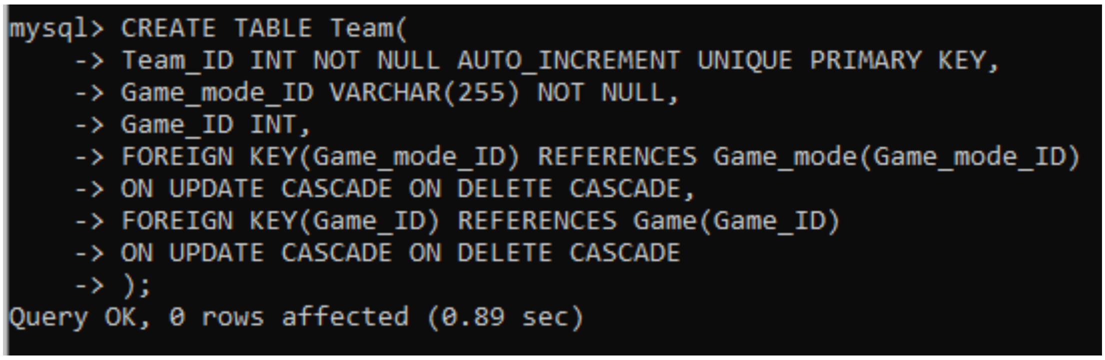
```
mysql> CREATE TABLE Mode_chosen(
    -> Player_ID INT NOT NULL,
    -> Game_mode_ID VARCHAR(255) NOT NULL,
    -> FOREIGN KEY (Player_ID) REFERENCES Player(Player_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE,
    -> FOREIGN KEY (Game_mode_ID) REFERENCES Game_mode(Game_mode_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.42 sec)
```

7. **Game_mode table:**



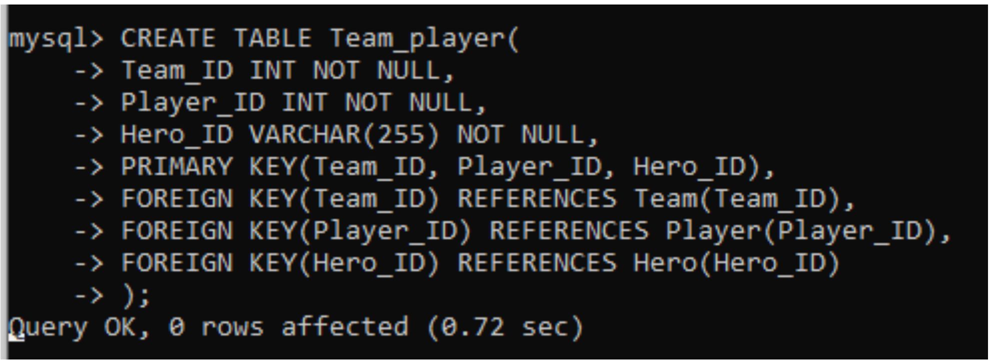
```
mysql> CREATE TABLE Game_mode(
    -> Game_mode_ID VARCHAR(255) NOT NULL PRIMARY KEY,
    -> Mode_type VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.24 sec)
```

8. **Team table:**



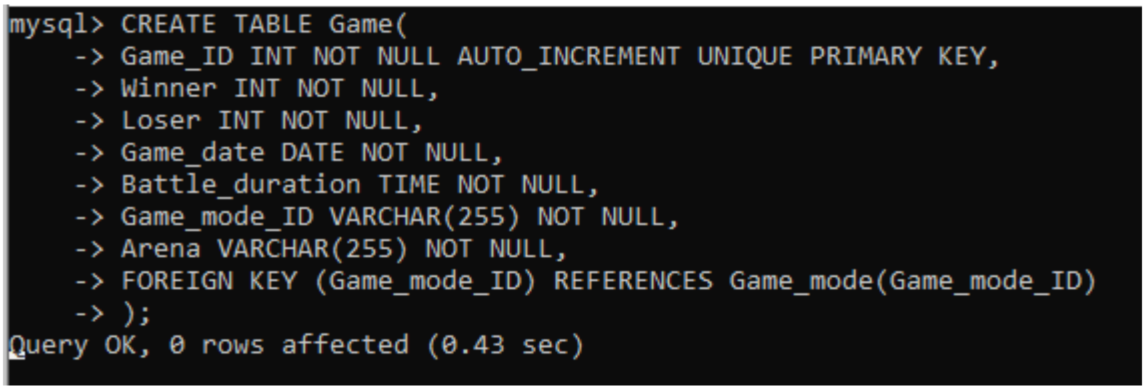
```
mysql> CREATE TABLE Team(
    -> Team_ID INT NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY,
    -> Game_mode_ID VARCHAR(255) NOT NULL,
    -> Game_ID INT,
    -> FOREIGN KEY(Game_mode_ID) REFERENCES Game_mode(Game_mode_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE,
    -> FOREIGN KEY(Game_ID) REFERENCES Game(Game_ID)
    -> ON UPDATE CASCADE ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.89 sec)
```

9. **Team_player table:**



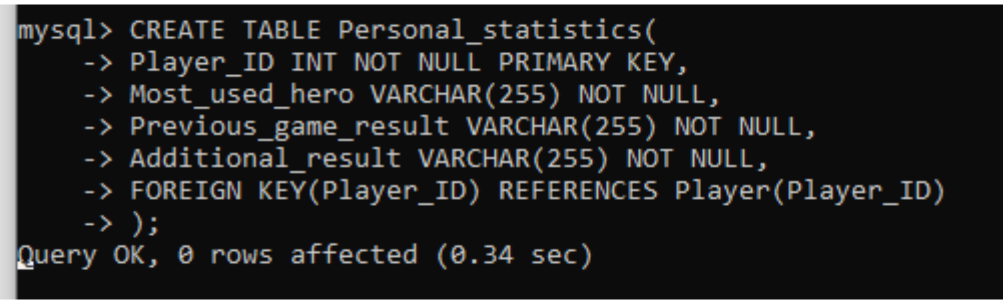
```
mysql> CREATE TABLE Team_player(
    -> Team_ID INT NOT NULL,
    -> Player_ID INT NOT NULL,
    -> Hero_ID VARCHAR(255) NOT NULL,
    -> PRIMARY KEY(Team_ID, Player_ID, Hero_ID),
    -> FOREIGN KEY(Team_ID) REFERENCES Team(Team_ID),
    -> FOREIGN KEY(Player_ID) REFERENCES Player(Player_ID),
    -> FOREIGN KEY(Hero_ID) REFERENCES Hero(Hero_ID)
    -> );
Query OK, 0 rows affected (0.72 sec)
```

10. **Game table:**



```
mysql> CREATE TABLE Game(
    -> Game_ID INT NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY,
    -> Winner INT NOT NULL,
    -> Loser INT NOT NULL,
    -> Game_date DATE NOT NULL,
    -> Battle_duration TIME NOT NULL,
    -> Game_mode_ID VARCHAR(255) NOT NULL,
    -> Arena VARCHAR(255) NOT NULL,
    -> FOREIGN KEY (Game_mode_ID) REFERENCES Game_mode(Game_mode_ID)
    -> );
Query OK, 0 rows affected (0.43 sec)
```

11. **Personal_statistics table:q**



```
mysql> CREATE TABLE Personal_statistics(
    -> Player_ID INT NOT NULL PRIMARY KEY,
    -> Most_used_hero VARCHAR(255) NOT NULL,
    -> Previous_game_result VARCHAR(255) NOT NULL,
    -> Additional_result VARCHAR(255) NOT NULL,
    -> FOREIGN KEY(Player_ID) REFERENCES Player(Player_ID)
    -> );
Query OK, 0 rows affected (0.34 sec)
```

Similarly, to update and delete data in child table automatically when the data are updated and deleted in parent table we need to add **ON UPDATE CASCADE** and **ON DELETE CASCADE** in the create_table syntax, respectively.

AUTO_INCREMETN: This command is used in order to generate unique number automatically when new data are inserted into the table.

Insert data in table:

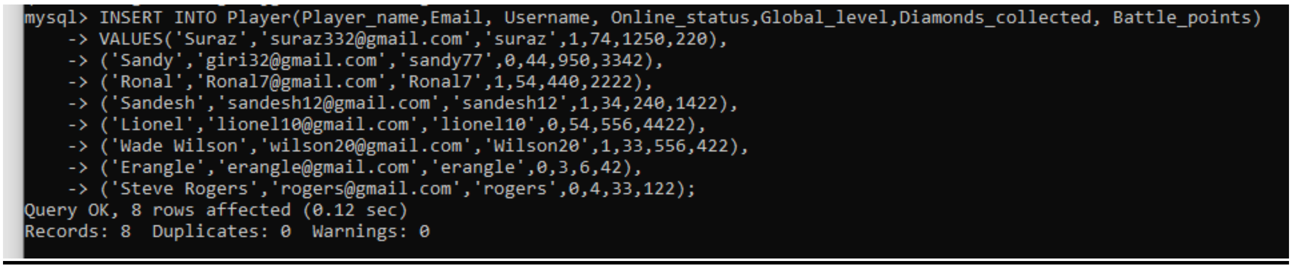
The syntax in order to insert the data in the table is:

INSERT INTO table_name(Column_1, Column_2, Column_3,…)

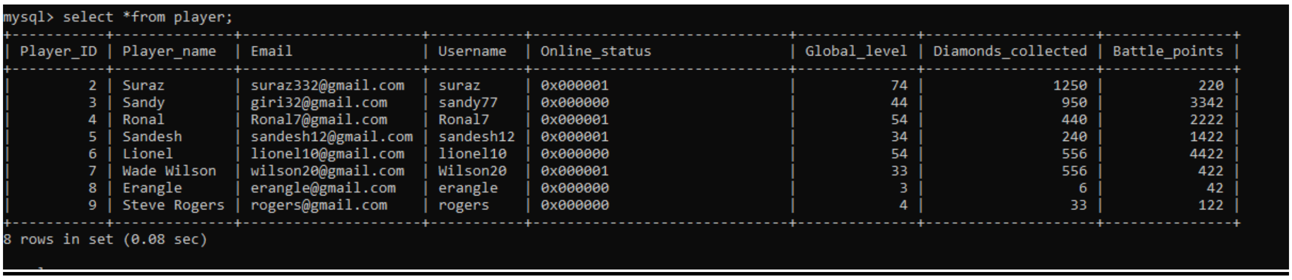
VALUES (Value_1, value_2, value_3, ….);

To insert the varchar or string values, we need to put the values inside inverted commas.

1. **<u>Player_table:</u>**

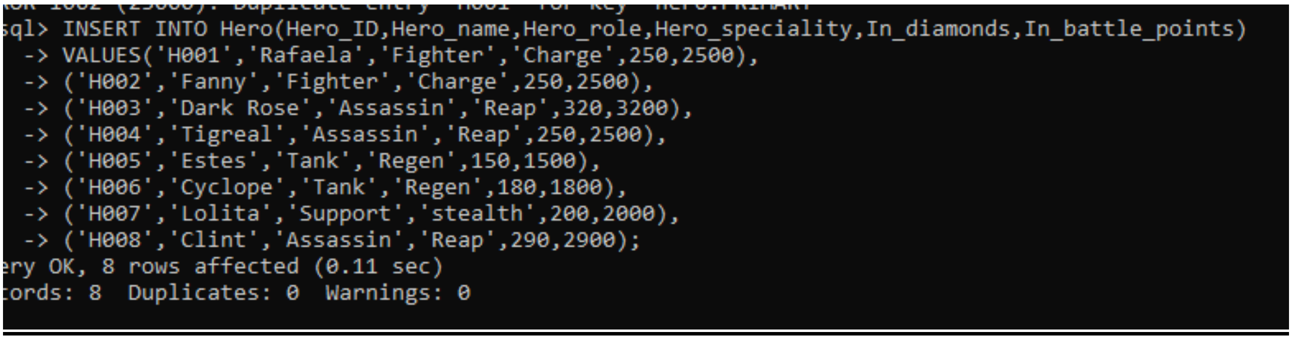


```
mysql> INSERT INTO Player(Player_name,Email, Username, Online_status,Global_level,Diamonds_collected, Battle_points)
    -> VALUES('Suraz','suraz332@gmail.com','suraz',1,74,1250,220),
    -> ('Sandy','giri32@gmail.com','sandy77',0,44,950,3342),
    -> ('Ronal','Ronal7@gmail.com','Ronal7',1,54,440,2222),
    -> ('Sandesh','sandesh12@gmail.com','sandesh12',1,34,240,1422),
    -> ('Lionel','lionel10@gmail.com','lionel10',0,54,556,4422),
    -> ('Wade Wilson','wilson20@gmail.com','Wilson20',1,33,556,422),
    -> ('Erangle','erangle@gmail.com','erangle',0,3,6,42),
    -> ('Steve Rogers','rogers@gmail.com','rogers',0,4,33,122);
Query OK, 8 rows affected (0.12 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

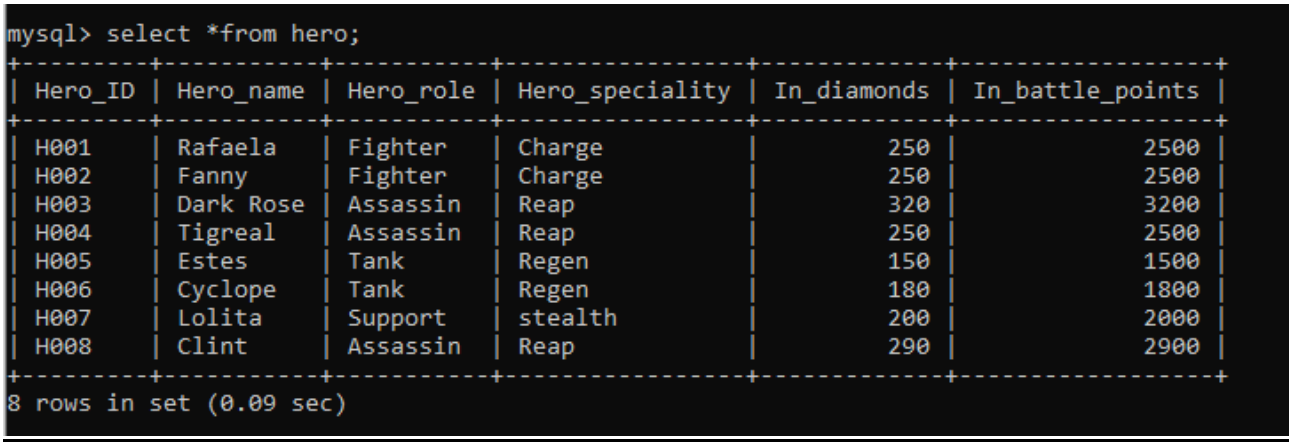


```
mysql> select *from player;
+-----------+--------------+---------------------+-----------+--------------------------------+--------------+--------------------+---------------+
| Player_ID | Player_name  | Email               | Username  | Online_status                  | Global_level | Diamonds_collected | Battle_points |
+-----------+--------------+---------------------+-----------+--------------------------------+--------------+--------------------+---------------+
|         2 | Suraz        | suraz332@gmail.com  | suraz     | 0x000001                       |           74 |               1250 |           220 |
|         3 | Sandy        | giri32@gmail.com    | sandy77   | 0x000000                       |           44 |                950 |          3342 |
|         4 | Ronal        | Ronal7@gmail.com    | Ronal7    | 0x000001                       |           54 |                440 |          2222 |
|         5 | Sandesh      | sandesh12@gmail.com | sandesh12 | 0x000001                       |           34 |                240 |          1422 |
|         6 | Lionel       | lionel10@gmail.com  | lionel10  | 0x000000                       |           54 |                556 |          4422 |
|         7 | Wade Wilson  | wilson20@gmail.com  | Wilson20  | 0x000001                       |           33 |                556 |           422 |
|         8 | Erangle      | erangle@gmail.com   | erangle   | 0x000000                       |            3 |                  6 |            42 |
|         9 | Steve Rogers | rogers@gmail.com    | rogers    | 0x000000                       |            4 |                 33 |           122 |
+-----------+--------------+---------------------+-----------+--------------------------------+--------------+--------------------+---------------+
8 rows in set (0.08 sec)
```

2. **<u>Hero_table:</u>**

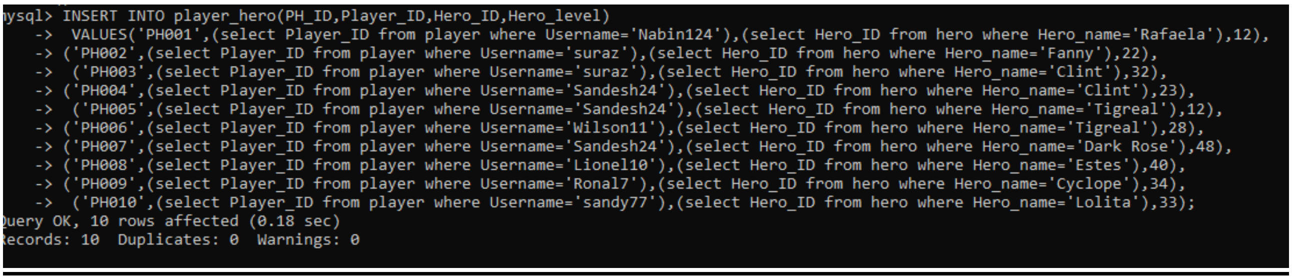


```
ROR 1062 (23000): Duplicate entry 'H001' for key 'hero.PRIMARY'
sql> INSERT INTO Hero(Hero_ID,Hero_name,Hero_role,Hero_speciality,In_diamonds,In_battle_points)
  -> VALUES('H001','Rafaela','Fighter','Charge',250,2500),
  -> ('H002','Fanny','Fighter','Charge',250,2500),
  -> ('H003','Dark Rose','Assassin','Reap',320,3200),
  -> ('H004','Tigreal','Assassin','Reap',250,2500),
  -> ('H005','Estes','Tank','Regen',150,1500),
  -> ('H006','Cyclope','Tank','Regen',180,1800),
  -> ('H007','Lolita','Support','stealth',200,2000),
  -> ('H008','Clint','Assassin','Reap',290,2900);
ery OK, 8 rows affected (0.11 sec)
cords: 8  Duplicates: 0  Warnings: 0
```

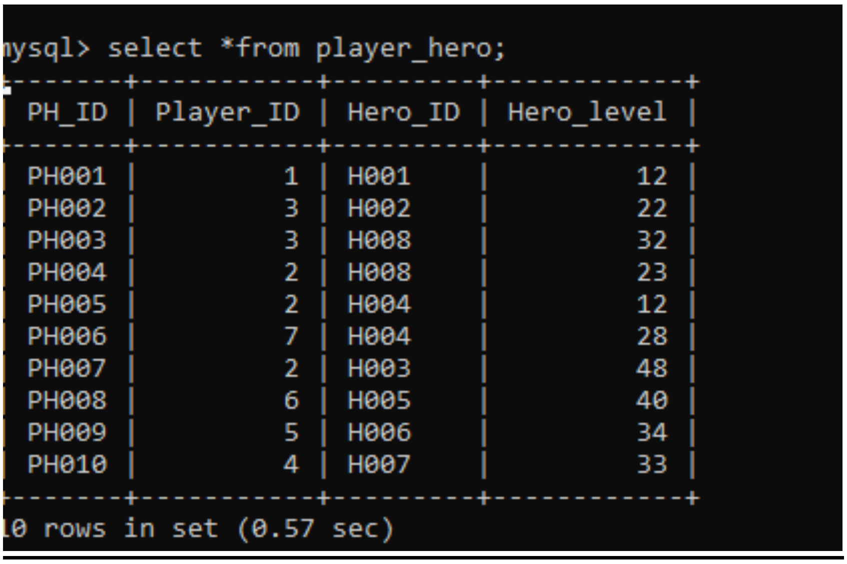


```
mysql> select *from hero;
+---------+-----------+----------+-----------------+-------------+------------------+
| Hero_ID | Hero_name | Hero_role | Hero_speciality | In_diamonds | In_battle_points |
+---------+-----------+----------+-----------------+-------------+------------------+
| H001    | Rafaela   | Fighter  | Charge          |         250 |             2500 |
| H002    | Fanny     | Fighter  | Charge          |         250 |             2500 |
| H003    | Dark Rose | Assassin | Reap            |         320 |             3200 |
| H004    | Tigreal   | Assassin | Reap            |         250 |             2500 |
| H005    | Estes     | Tank     | Regen           |         150 |             1500 |
| H006    | Cyclope   | Tank     | Regen           |         180 |             1800 |
| H007    | Lolita    | Support  | stealth         |         200 |             2000 |
| H008    | Clint     | Assassin | Reap            |         290 |             2900 |
+---------+-----------+----------+-----------------+-------------+------------------+
8 rows in set (0.09 sec)
```

3. Player_hero table:

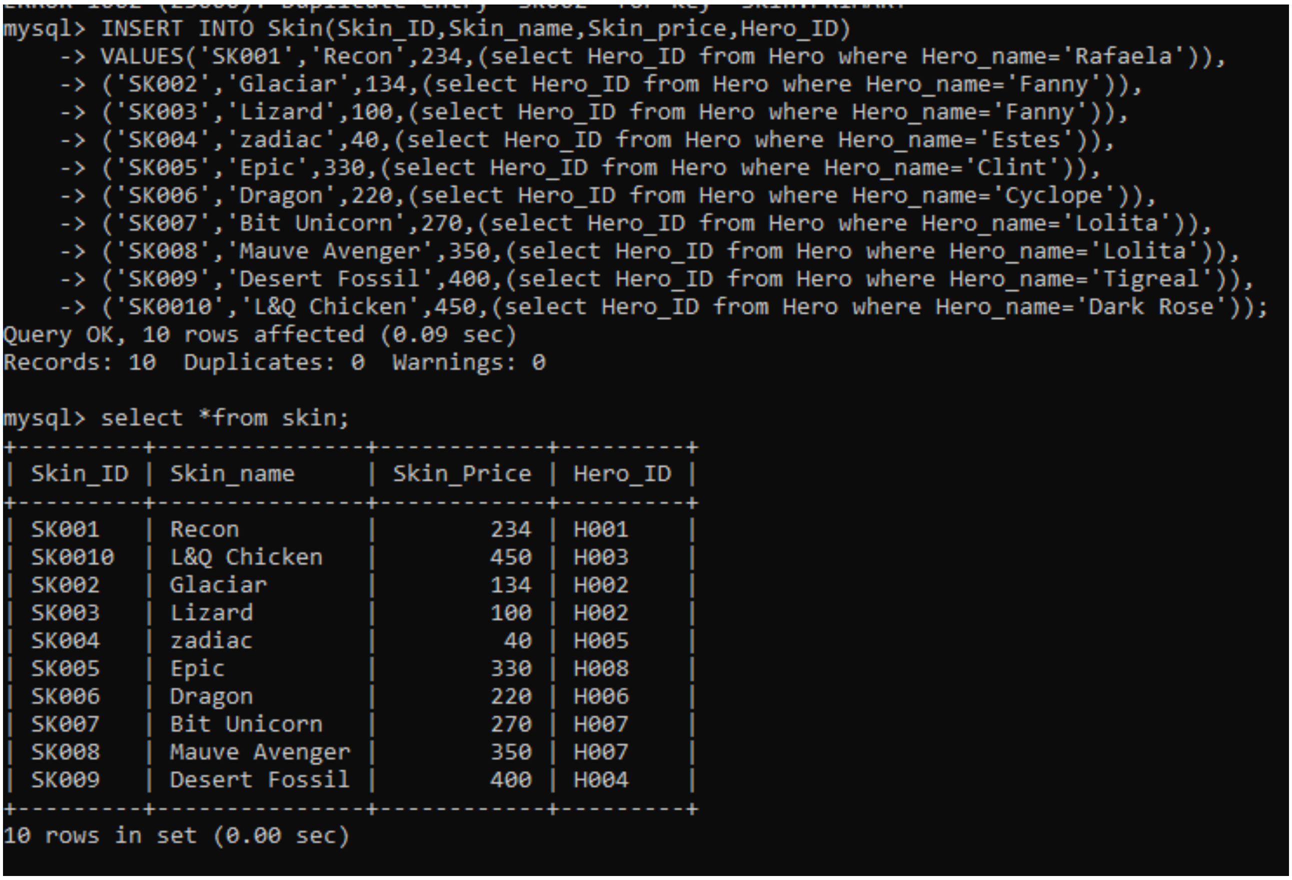


```
ysql> INSERT INTO player_hero(PH_ID,Player_ID,Hero_ID,Hero_level)
    ->  VALUES('PH001',(select Player_ID from player where Username='Nabin124'),(select Hero_ID from hero where Hero_name='Rafaela'),12),
    -> ('PH002',(select Player_ID from player where Username='suraz'),(select Hero_ID from hero where Hero_name='Fanny'),22),
    ->  ('PH003',(select Player_ID from player where Username='suraz'),(select Hero_ID from hero where Hero_name='Clint'),32),
    -> ('PH004',(select Player_ID from player where Username='Sandesh24'),(select Hero_ID from hero where Hero_name='Clint'),23),
    ->  ('PH005',(select Player_ID from player where Username='Sandesh24'),(select Hero_ID from hero where Hero_name='Tigreal'),12),
    -> ('PH006',(select Player_ID from player where Username='Wilson11'),(select Hero_ID from hero where Hero_name='Tigreal'),28),
    -> ('PH007',(select Player_ID from player where Username='Sandesh24'),(select Hero_ID from hero where Hero_name='Dark Rose'),48),
    -> ('PH008',(select Player_ID from player where Username='Lionel10'),(select Hero_ID from hero where Hero_name='Estes'),40),
    -> ('PH009',(select Player_ID from player where Username='Ronal7'),(select Hero_ID from hero where Hero_name='Cyclope'),34),
    ->  ('PH010',(select Player_ID from player where Username='sandy77'),(select Hero_ID from hero where Hero_name='Lolita'),33);
uery OK, 10 rows affected (0.18 sec)
ecords: 10  Duplicates: 0  Warnings: 0
```



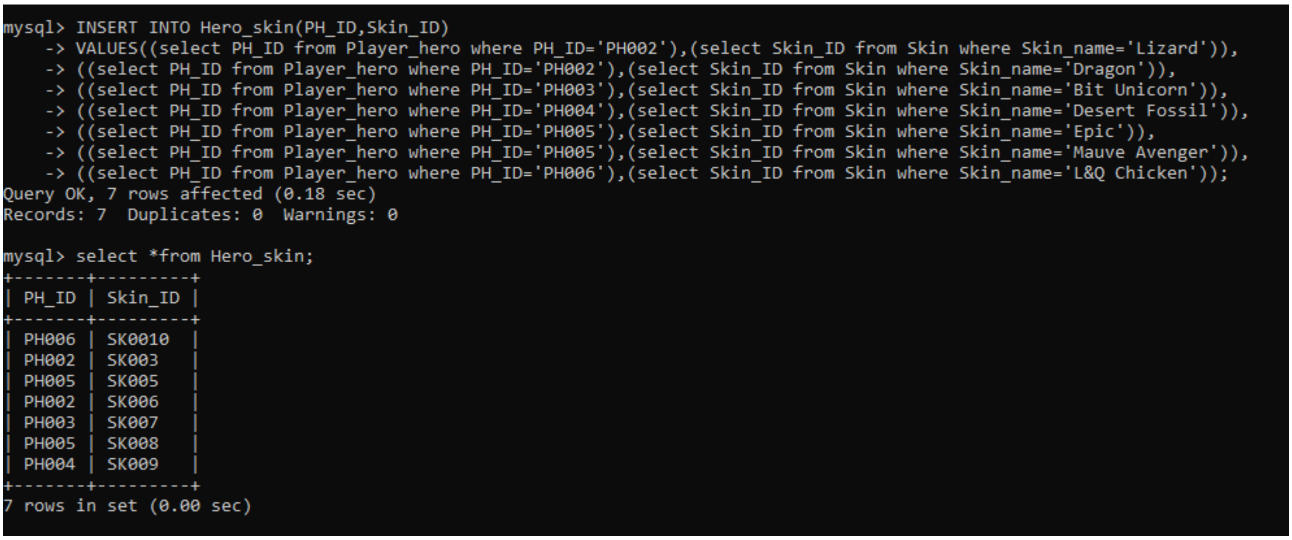
```
ysql> select *from player_hero;
-------+-----------+---------+------------+
 PH_ID | Player_ID | Hero_ID | Hero_level |
-------+-----------+---------+------------+
 PH001 |         1 | H001    |         12 |
 PH002 |         3 | H002    |         22 |
 PH003 |         3 | H008    |         32 |
 PH004 |         2 | H008    |         23 |
 PH005 |         2 | H004    |         12 |
 PH006 |         7 | H004    |         28 |
 PH007 |         2 | H003    |         48 |
 PH008 |         6 | H005    |         40 |
 PH009 |         5 | H006    |         34 |
 PH010 |         4 | H007    |         33 |
-------+-----------+---------+------------+
0 rows in set (0.57 sec)
```

4. Skin_table:



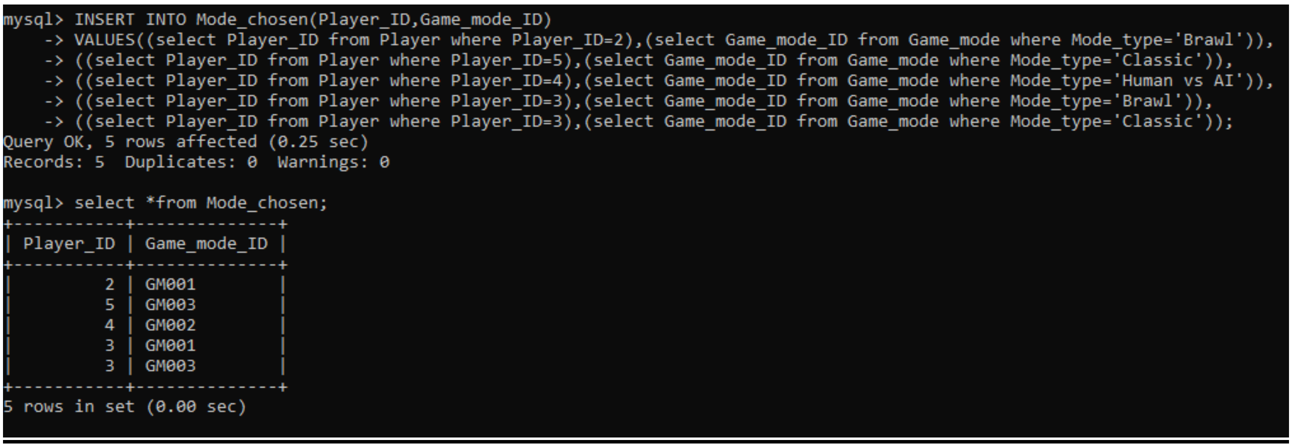
```
mysql> INSERT INTO Skin(Skin_ID,Skin_name,Skin_price,Hero_ID)
    -> VALUES('SK001','Recon',234,(select Hero_ID from Hero where Hero_name='Rafaela')),
    -> ('SK002','Glaciar',134,(select Hero_ID from Hero where Hero_name='Fanny')),
    -> ('SK003','Lizard',100,(select Hero_ID from Hero where Hero_name='Fanny')),
    -> ('SK004','zadiac',40,(select Hero_ID from Hero where Hero_name='Estes')),
    -> ('SK005','Epic',330,(select Hero_ID from Hero where Hero_name='Clint')),
    -> ('SK006','Dragon',220,(select Hero_ID from Hero where Hero_name='Cyclope')),
    -> ('SK007','Bit Unicorn',270,(select Hero_ID from Hero where Hero_name='Lolita')),
    -> ('SK008','Mauve Avenger',350,(select Hero_ID from Hero where Hero_name='Lolita')),
    -> ('SK009','Desert Fossil',400,(select Hero_ID from Hero where Hero_name='Tigreal')),
    -> ('SK0010','L&Q Chicken',450,(select Hero_ID from Hero where Hero_name='Dark Rose'));
Query OK, 10 rows affected (0.09 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select *from skin;
+---------+---------------+------------+---------+
| Skin_ID | Skin_name     | Skin_Price | Hero_ID |
+---------+---------------+------------+---------+
| SK001   | Recon         |        234 | H001    |
| SK0010  | L&Q Chicken   |        450 | H003    |
| SK002   | Glaciar       |        134 | H002    |
| SK003   | Lizard        |        100 | H002    |
| SK004   | zadiac        |         40 | H005    |
| SK005   | Epic          |        330 | H008    |
| SK006   | Dragon        |        220 | H006    |
| SK007   | Bit Unicorn   |        270 | H007    |
| SK008   | Mauve Avenger |        350 | H007    |
| SK009   | Desert Fossil |        400 | H004    |
+---------+---------------+------------+---------+
10 rows in set (0.00 sec)
```

5. Hero_skin table:



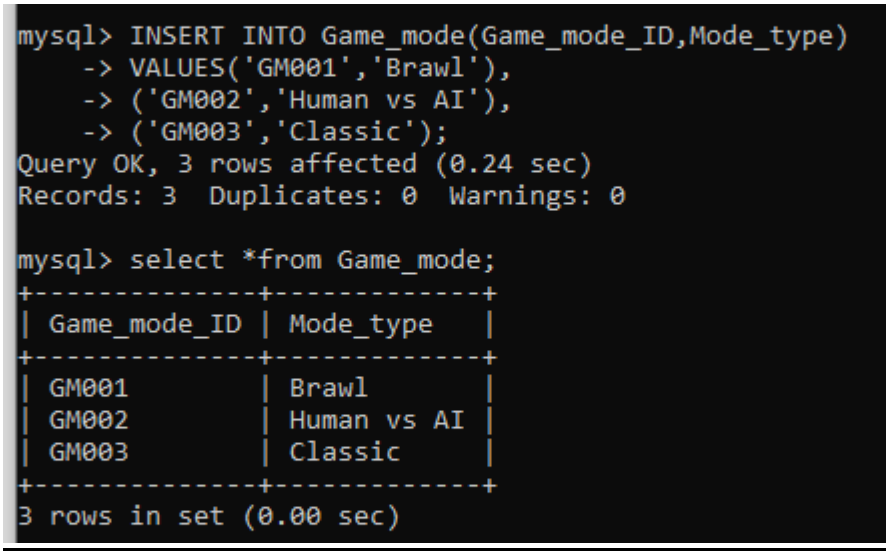
```
mysql> INSERT INTO Hero_skin(PH_ID,Skin_ID)
    -> VALUES((select PH_ID from Player_hero where PH_ID='PH002'),(select Skin_ID from Skin where Skin_name='Lizard')),
    -> ((select PH_ID from Player_hero where PH_ID='PH002'),(select Skin_ID from Skin where Skin_name='Dragon')),
    -> ((select PH_ID from Player_hero where PH_ID='PH003'),(select Skin_ID from Skin where Skin_name='Bit Unicorn')),
    -> ((select PH_ID from Player_hero where PH_ID='PH004'),(select Skin_ID from Skin where Skin_name='Desert Fossil')),
    -> ((select PH_ID from Player_hero where PH_ID='PH005'),(select Skin_ID from Skin where Skin_name='Epic')),
    -> ((select PH_ID from Player_hero where PH_ID='PH005'),(select Skin_ID from Skin where Skin_name='Mauve Avenger')),
    -> ((select PH_ID from Player_hero where PH_ID='PH006'),(select Skin_ID from Skin where Skin_name='L&Q Chicken'));
Query OK, 7 rows affected (0.18 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> select *from Hero_skin;
+-------+---------+
| PH_ID | Skin_ID |
+-------+---------+
| PH006 | SK0010  |
| PH002 | SK003   |
| PH005 | SK005   |
| PH002 | SK006   |
| PH003 | SK007   |
| PH005 | SK008   |
| PH004 | SK009   |
+-------+---------+
7 rows in set (0.00 sec)
```

6. **Mode_chosen table:**



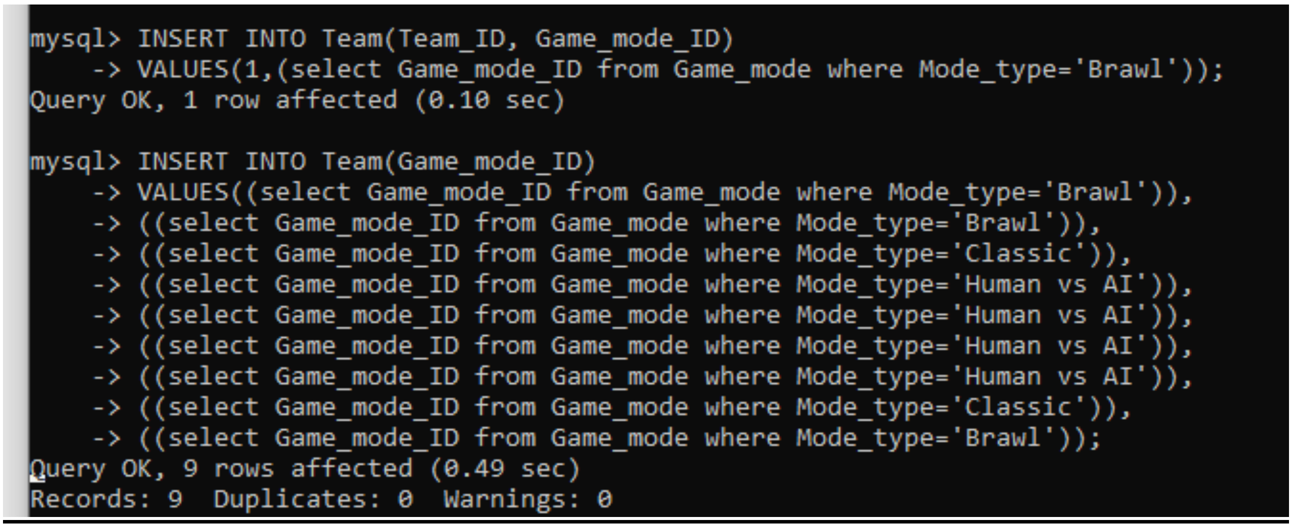
```
mysql> INSERT INTO Mode_chosen(Player_ID,Game_mode_ID)
    -> VALUES((select Player_ID from Player where Player_ID=2),(select Game_mode_ID from Game_mode where Mode_type='Brawl')),
    -> ((select Player_ID from Player where Player_ID=5),(select Game_mode_ID from Game_mode where Mode_type='Classic')),
    -> ((select Player_ID from Player where Player_ID=4),(select Game_mode_ID from Game_mode where Mode_type='Human vs AI')),
    -> ((select Player_ID from Player where Player_ID=3),(select Game_mode_ID from Game_mode where Mode_type='Brawl')),
    -> ((select Player_ID from Player where Player_ID=3),(select Game_mode_ID from Game_mode where Mode_type='Classic'));
Query OK, 5 rows affected (0.25 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select *from Mode_chosen;
+-----------+--------------+
| Player_ID | Game_mode_ID |
+-----------+--------------+
|         2 | GM001        |
|         5 | GM003        |
|         4 | GM002        |
|         3 | GM001        |
|         3 | GM003        |
+-----------+--------------+
5 rows in set (0.00 sec)
```

7. **Game_mode table:**

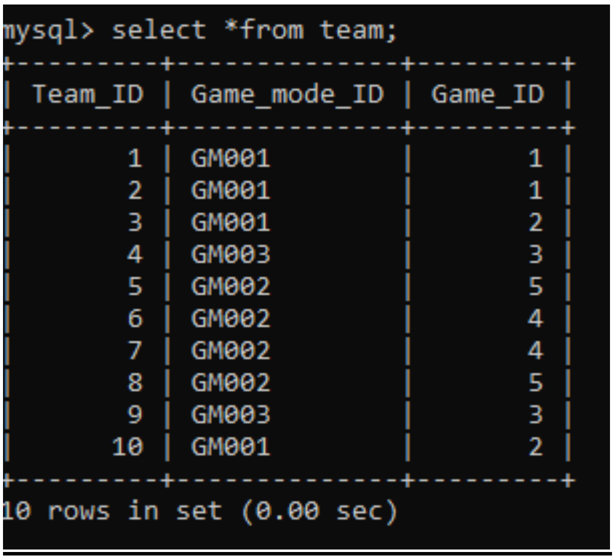


```
mysql> INSERT INTO Game_mode(Game_mode_ID,Mode_type)
    -> VALUES('GM001','Brawl'),
    -> ('GM002','Human vs AI'),
    -> ('GM003','Classic');
Query OK, 3 rows affected (0.24 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select *from Game_mode;
+--------------+-------------+
| Game_mode_ID | Mode_type   |
+--------------+-------------+
| GM001        | Brawl       |
| GM002        | Human vs AI |
| GM003        | Classic     |
+--------------+-------------+
3 rows in set (0.00 sec)
```

8. **Team_table:**

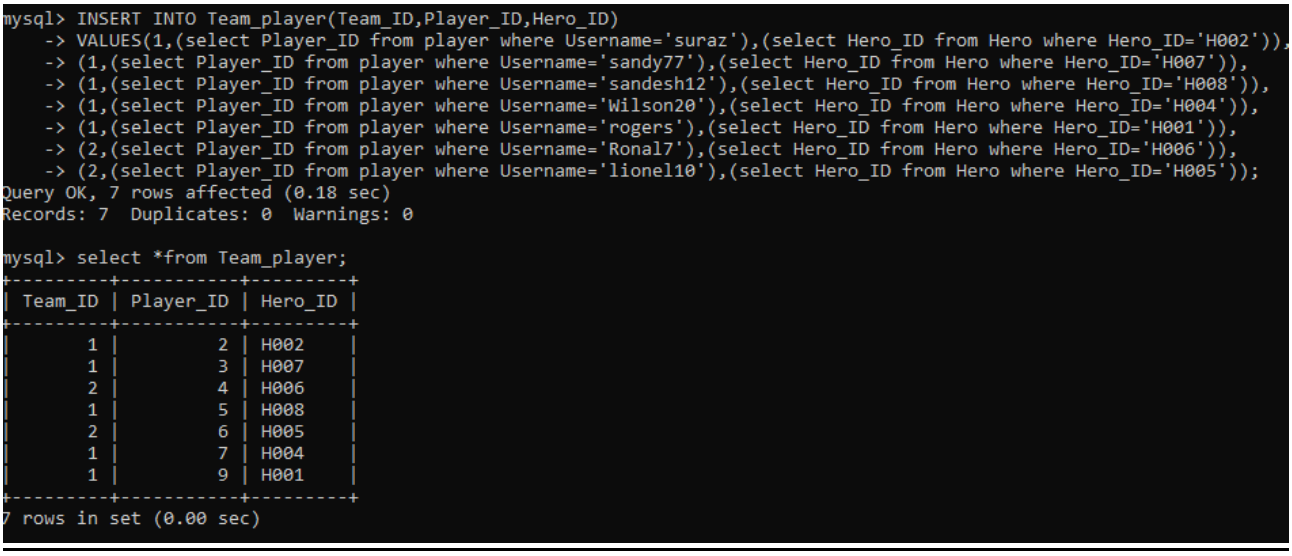


```
mysql> INSERT INTO Team(Team_ID, Game_mode_ID)
    -> VALUES(1,(select Game_mode_ID from Game_mode where Mode_type='Brawl'));
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO Team(Game_mode_ID)
    -> VALUES((select Game_mode_ID from Game_mode where Mode_type='Brawl')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Brawl')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Classic')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Human vs AI')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Human vs AI')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Human vs AI')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Human vs AI')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Classic')),
    -> ((select Game_mode_ID from Game_mode where Mode_type='Brawl'));
Query OK, 9 rows affected (0.49 sec)
Records: 9  Duplicates: 0  Warnings: 0
```



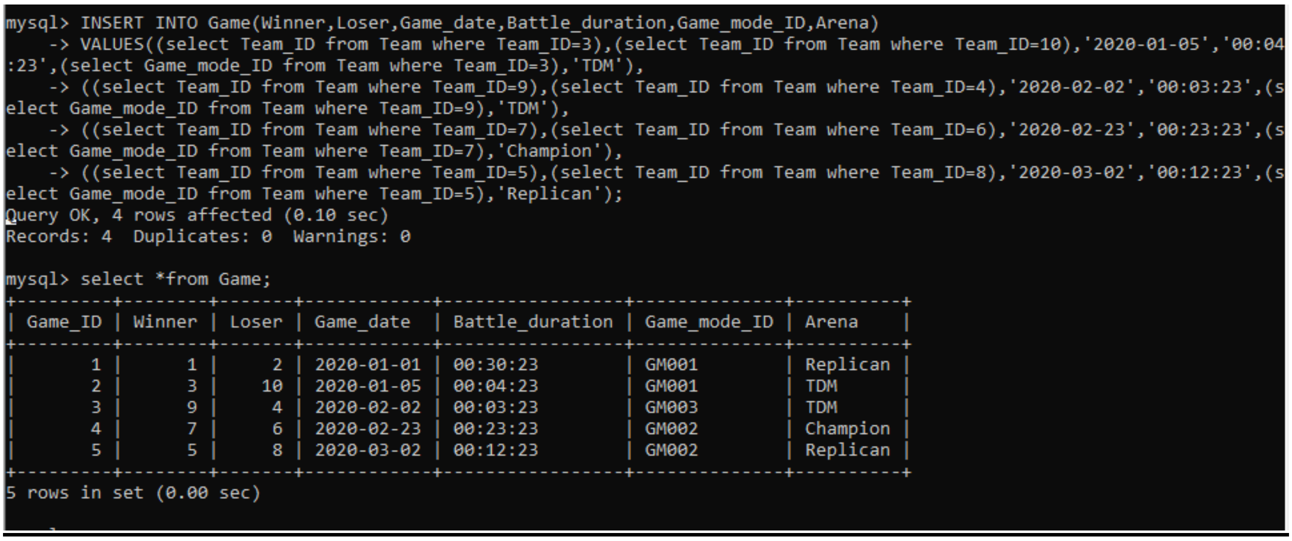
```
mysql> select *from team;
+---------+--------------+---------+
| Team_ID | Game_mode_ID | Game_ID |
+---------+--------------+---------+
|       1 | GM001        |       1 |
|       2 | GM001        |       1 |
|       3 | GM001        |       2 |
|       4 | GM003        |       3 |
|       5 | GM002        |       5 |
|       6 | GM002        |       4 |
|       7 | GM002        |       4 |
|       8 | GM002        |       5 |
|       9 | GM003        |       3 |
|      10 | GM001        |       2 |
+---------+--------------+---------+
10 rows in set (0.00 sec)
```

9. **Team_player table:**



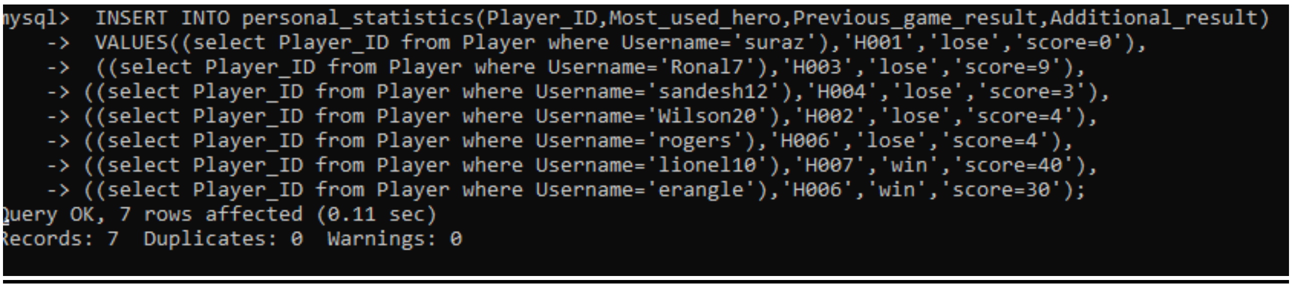
```
mysql> INSERT INTO Team_player(Team_ID,Player_ID,Hero_ID)
    -> VALUES(1,(select Player_ID from player where Username='suraz'),(select Hero_ID from Hero where Hero_ID='H002')),
    -> (1,(select Player_ID from player where Username='sandy77'),(select Hero_ID from Hero where Hero_ID='H007')),
    -> (1,(select Player_ID from player where Username='sandesh12'),(select Hero_ID from Hero where Hero_ID='H008')),
    -> (1,(select Player_ID from player where Username='Wilson20'),(select Hero_ID from Hero where Hero_ID='H004')),
    -> (1,(select Player_ID from player where Username='rogers'),(select Hero_ID from Hero where Hero_ID='H001')),
    -> (2,(select Player_ID from player where Username='Ronal7'),(select Hero_ID from Hero where Hero_ID='H006')),
    -> (2,(select Player_ID from player where Username='lionel10'),(select Hero_ID from Hero where Hero_ID='H005'));
Query OK, 7 rows affected (0.18 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> select *from Team_player;
+---------+-----------+---------+
| Team_ID | Player_ID | Hero_ID |
+---------+-----------+---------+
|       1 |         2 | H002    |
|       1 |         3 | H007    |
|       2 |         4 | H006    |
|       1 |         5 | H008    |
|       2 |         6 | H005    |
|       1 |         7 | H004    |
|       1 |         9 | H001    |
+---------+-----------+---------+
7 rows in set (0.00 sec)
```

10. Game table:

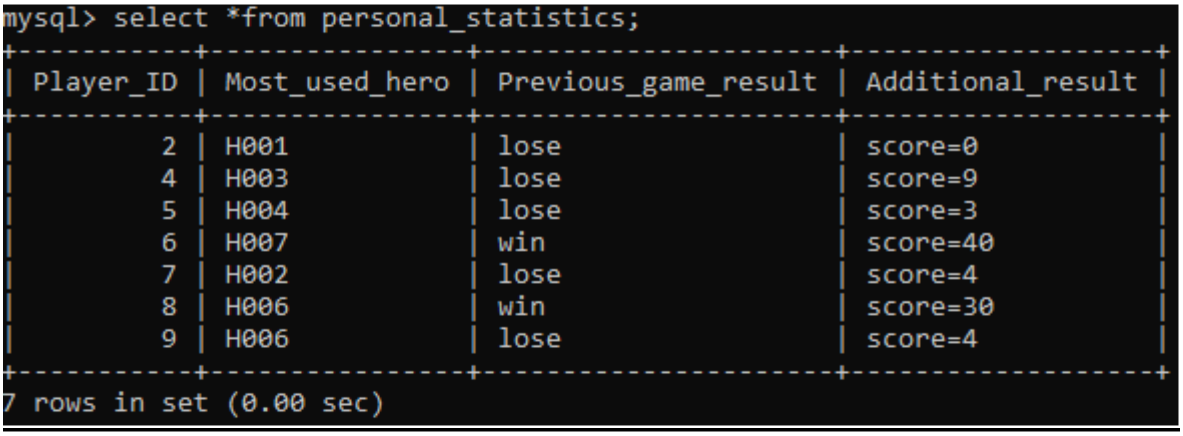


```
mysql> INSERT INTO Game(Winner,Loser,Game_date,Battle_duration,Game_mode_ID,Arena)
    -> VALUES((select Team_ID from Team where Team_ID=3),(select Team_ID from Team where Team_ID=10),'2020-01-05','00:04
:23',(select Game_mode_ID from Team where Team_ID=3),'TDM'),
    -> ((select Team_ID from Team where Team_ID=9),(select Team_ID from Team where Team_ID=4),'2020-02-02','00:03:23',(s
elect Game_mode_ID from Team where Team_ID=9),'TDM'),
    -> ((select Team_ID from Team where Team_ID=7),(select Team_ID from Team where Team_ID=6),'2020-02-23','00:23:23',(s
elect Game_mode_ID from Team where Team_ID=7),'Champion'),
    -> ((select Team_ID from Team where Team_ID=5),(select Team_ID from Team where Team_ID=8),'2020-03-02','00:12:23',(s
elect Game_mode_ID from Team where Team_ID=5),'Replican');
Query OK, 4 rows affected (0.10 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select *from Game;
+---------+--------+-------+------------+-----------------+--------------+----------+
| Game_ID | Winner | Loser | Game_date  | Battle_duration | Game_mode_ID | Arena    |
+---------+--------+-------+------------+-----------------+--------------+----------+
|       1 |      1 |     2 | 2020-01-01 | 00:30:23        | GM001        | Replican |
|       2 |      3 |    10 | 2020-01-05 | 00:04:23        | GM001        | TDM      |
|       3 |      9 |     4 | 2020-02-02 | 00:03:23        | GM003        | TDM      |
|       4 |      7 |     6 | 2020-02-23 | 00:23:23        | GM002        | Champion |
|       5 |      5 |     8 | 2020-03-02 | 00:12:23        | GM002        | Replican |
+---------+--------+-------+------------+-----------------+--------------+----------+
5 rows in set (0.00 sec)
```

11. Personal_statistics table:



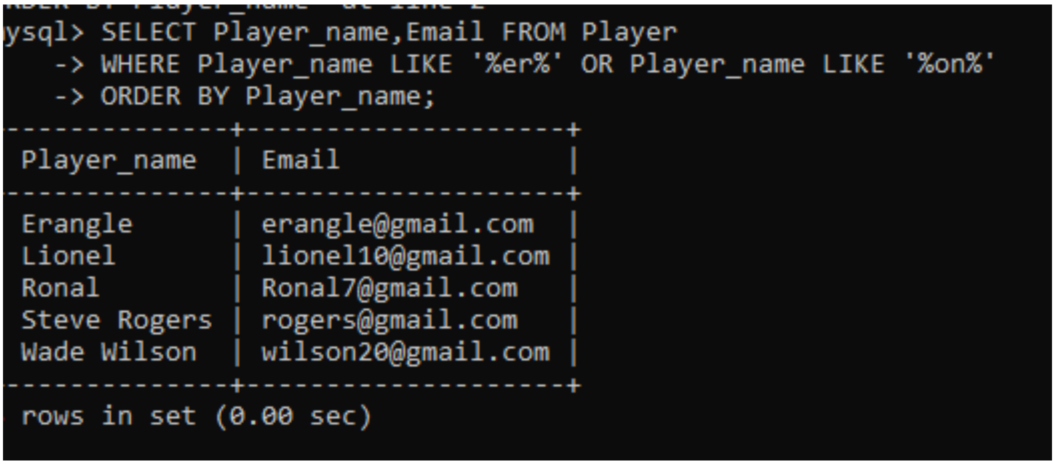
```
mysql>  INSERT INTO personal_statistics(Player_ID,Most_used_hero,Previous_game_result,Additional_result)
   ->   VALUES((select Player_ID from Player where Username='suraz'),'H001','lose','score=0'),
   ->   ((select Player_ID from Player where Username='Ronal7'),'H003','lose','score=9'),
   -> ((select Player_ID from Player where Username='sandesh12'),'H004','lose','score=3'),
   -> ((select Player_ID from Player where Username='Wilson20'),'H002','lose','score=4'),
   -> ((select Player_ID from Player where Username='rogers'),'H006','lose','score=4'),
   -> ((select Player_ID from Player where Username='lionel10'),'H007','win','score=40'),
   -> ((select Player_ID from Player where Username='erangle'),'H006','win','score=30');
Query OK, 7 rows affected (0.11 sec)
Records: 7  Duplicates: 0  Warnings: 0
```



```
mysql> select *from personal_statistics;
+-----------+----------------+----------------------+-------------------+
| Player_ID | Most_used_hero | Previous_game_result | Additional_result |
+-----------+----------------+----------------------+-------------------+
|         2 | H001           | lose                 | score=0           |
|         4 | H003           | lose                 | score=9           |
|         5 | H004           | lose                 | score=3           |
|         6 | H007           | win                  | score=40          |
|         7 | H002           | lose                 | score=4           |
|         8 | H006           | win                  | score=30          |
|         9 | H006           | lose                 | score=4           |
+-----------+----------------+----------------------+-------------------+
7 rows in set (0.00 sec)
```

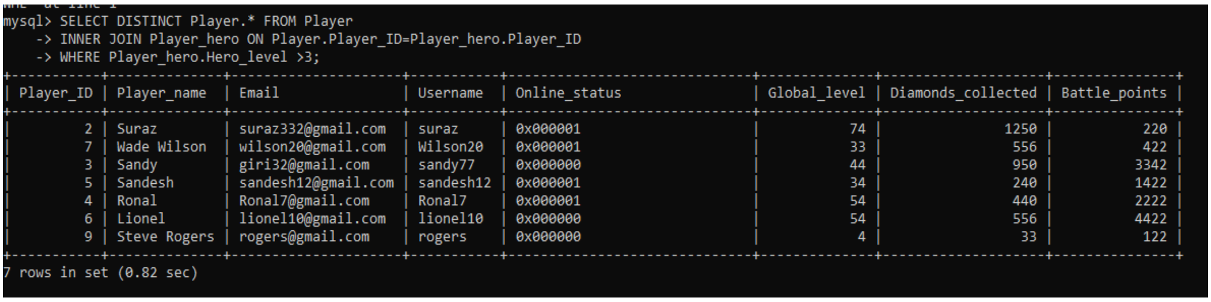
2. SQL-Data Manipulation Language (DML):

Question **1:**



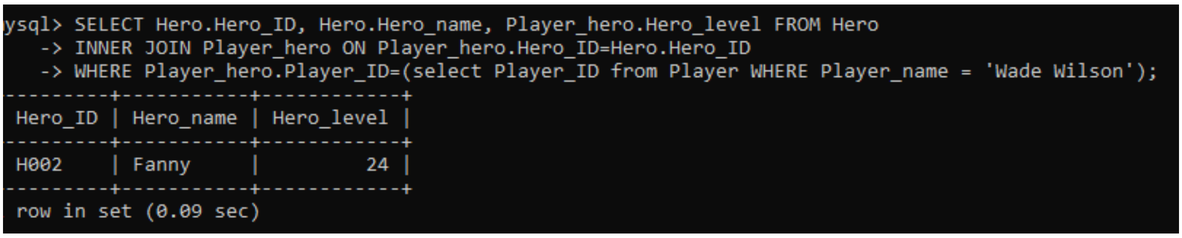
```
ysql> SELECT Player_name,Email FROM Player
    -> WHERE Player_name LIKE '%er%' OR Player_name LIKE '%on%'
    -> ORDER BY Player_name;
-------------+--------------------+
 Player_name  | Email              |
-------------+--------------------+
 Erangle      | erangle@gmail.com  |
 Lionel       | lionel10@gmail.com |
 Ronal        | Ronal7@gmail.com   |
 Steve Rogers | rogers@gmail.com   |
 Wade Wilson  | wilson20@gmail.com |
-------------+--------------------+
 rows in set (0.00 sec)
```

Question 2:



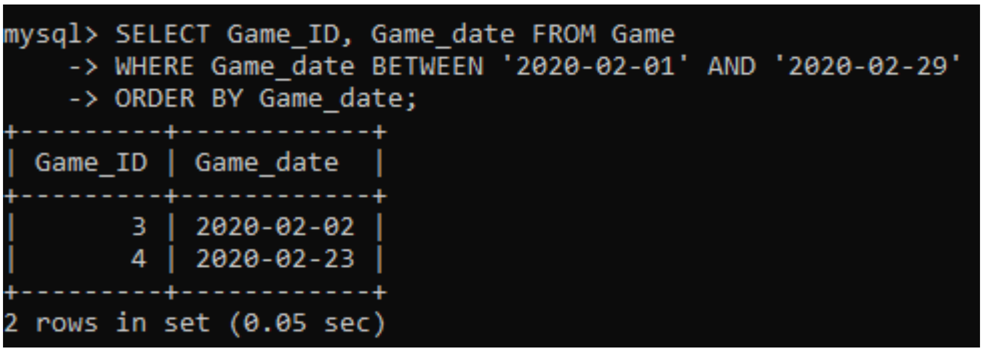
```
mysql> SELECT DISTINCT Player.* FROM Player
    -> INNER JOIN Player_hero ON Player.Player_ID=Player_hero.Player_ID
    -> WHERE Player_hero.Hero_level >3;
+-----------+--------------+---------------------+-----------+--------------------------+--------------+--------------------+--------------+
| Player_ID | Player_name  | Email               | Username  | Online_status            | Global_level | Diamonds_collected | Battle_points |
+-----------+--------------+---------------------+-----------+--------------------------+--------------+--------------------+--------------+
|         2 | Suraz        | suraz332@gmail.com  | suraz     | 0x000001                 |           74 |               1250 |          220 |
|         7 | Wade Wilson  | wilson20@gmail.com  | Wilson20  | 0x000001                 |           33 |                556 |          422 |
|         3 | Sandy        | giri32@gmail.com    | sandy77   | 0x000000                 |           44 |                950 |         3342 |
|         5 | Sandesh      | sandesh12@gmail.com | sandesh12 | 0x000001                 |           34 |                240 |         1422 |
|         4 | Ronal        | Ronal7@gmail.com    | Ronal7    | 0x000001                 |           54 |                440 |         2222 |
|         6 | Lionel       | lionel10@gmail.com  | lionel10  | 0x000000                 |           54 |                556 |         4422 |
|         9 | Steve Rogers | rogers@gmail.com    | rogers    | 0x000000                 |            4 |                 33 |          122 |
+-----------+--------------+---------------------+-----------+--------------------------+--------------+--------------------+--------------+
7 rows in set (0.82 sec)
```

Question 3:



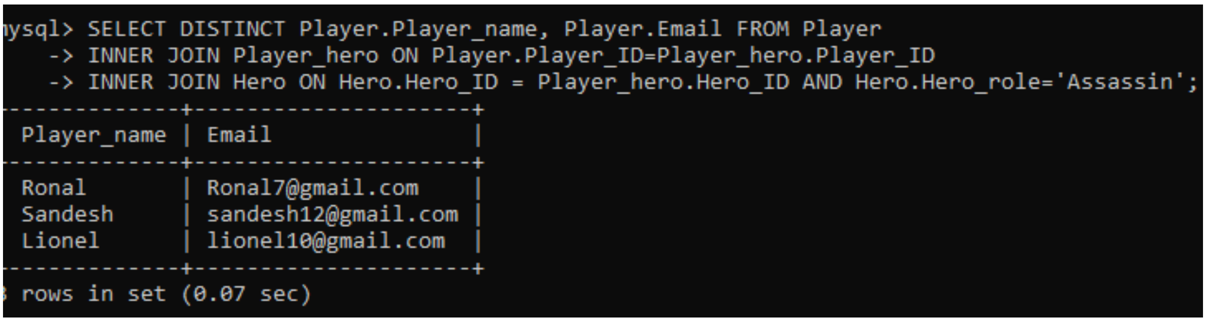
```
ysql> SELECT Hero.Hero_ID, Hero.Hero_name, Player_hero.Hero_level FROM Hero
    -> INNER JOIN Player_hero ON Player_hero.Hero_ID=Hero.Hero_ID
    -> WHERE Player_hero.Player_ID=(select Player_ID from Player WHERE Player_name = 'Wade Wilson');
--------+-----------+------------+
 Hero_ID | Hero_name | Hero_level |
--------+-----------+------------+
 H002    | Fanny     |         24 |
--------+-----------+------------+
 row in set (0.09 sec)
```

Question 4:



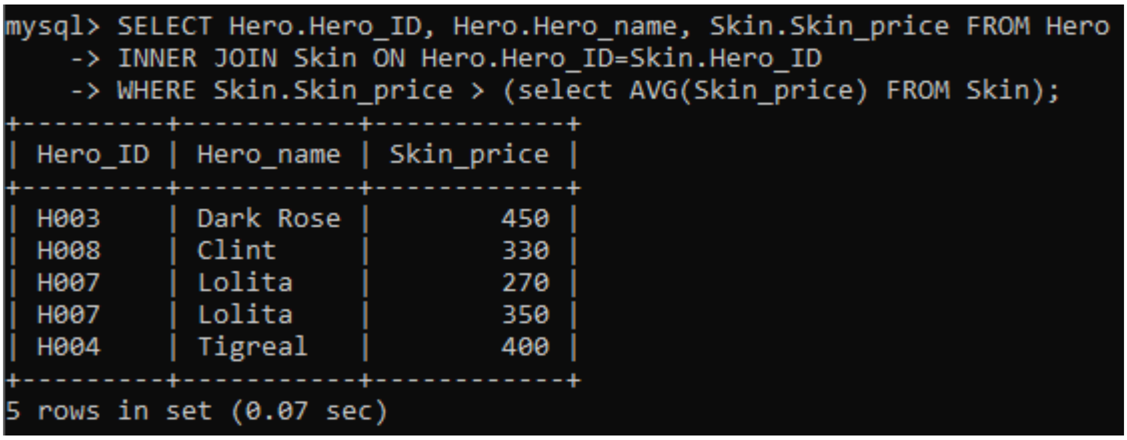
```
mysql> SELECT Game_ID, Game_date FROM Game
    -> WHERE Game_date BETWEEN '2020-02-01' AND '2020-02-29'
    -> ORDER BY Game_date;
+---------+------------+
| Game_ID | Game_date  |
+---------+------------+
|       3 | 2020-02-02 |
|       4 | 2020-02-23 |
+---------+------------+
2 rows in set (0.05 sec)
```

Question 5:



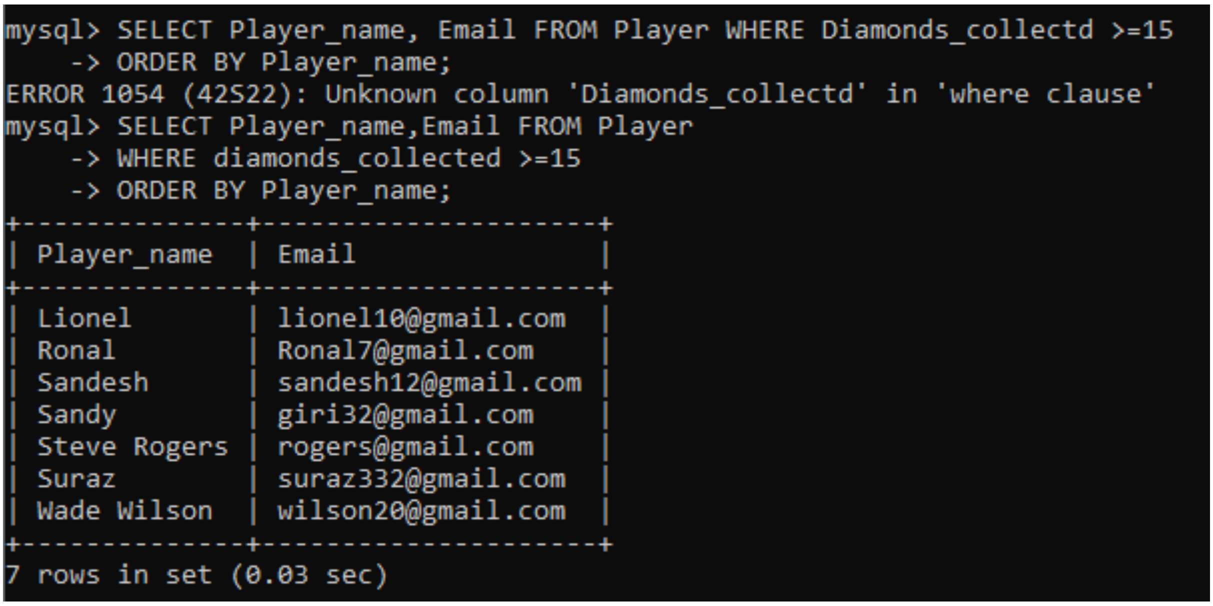
```
ysql> SELECT DISTINCT Player.Player_name, Player.Email FROM Player
    -> INNER JOIN Player_hero ON Player.Player_ID=Player_hero.Player_ID
    -> INNER JOIN Hero ON Hero.Hero_ID = Player_hero.Hero_ID AND Hero.Hero_role='Assassin';
------------+--------------------+
 Player_name | Email              |
------------+--------------------+
 Ronal       | Ronal7@gmail.com   |
 Sandesh     | sandesh12@gmail.com |
 Lionel      | lionel10@gmail.com |
------------+--------------------+
3 rows in set (0.07 sec)
```

Question 6:



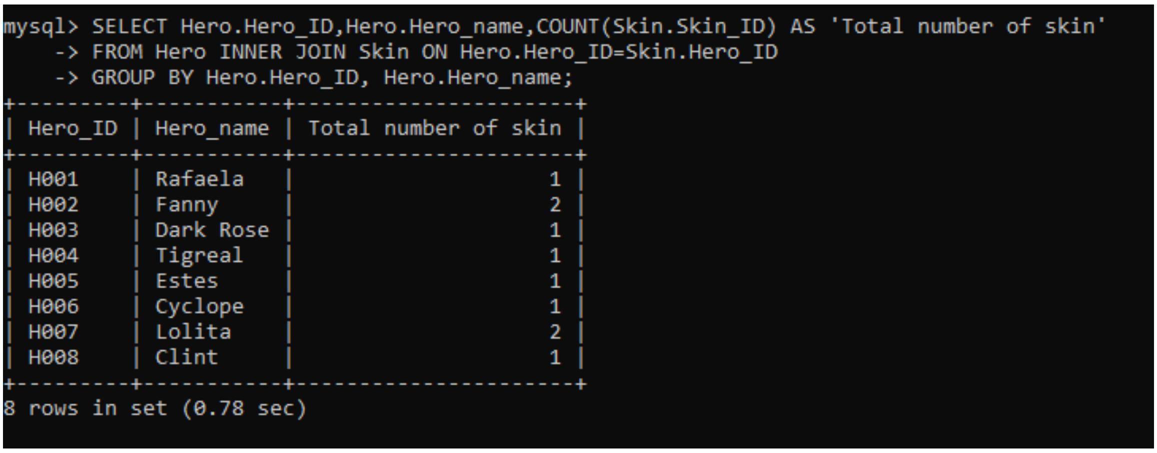
```
mysql> SELECT Hero.Hero_ID, Hero.Hero_name, Skin.Skin_price FROM Hero
    -> INNER JOIN Skin ON Hero.Hero_ID=Skin.Hero_ID
    -> WHERE Skin.Skin_price > (select AVG(Skin_price) FROM Skin);
+---------+-----------+------------+
| Hero_ID | Hero_name | Skin_price |
+---------+-----------+------------+
| H003    | Dark Rose |        450 |
| H008    | Clint     |        330 |
| H007    | Lolita    |        270 |
| H007    | Lolita    |        350 |
| H004    | Tigreal   |        400 |
+---------+-----------+------------+
5 rows in set (0.07 sec)
```

Question 7:



```
mysql> SELECT Player_name, Email FROM Player WHERE Diamonds_collectd >=15
    -> ORDER BY Player_name;
ERROR 1054 (42S22): Unknown column 'Diamonds_collectd' in 'where clause'
mysql> SELECT Player_name,Email FROM Player
    -> WHERE diamonds_collected >=15
    -> ORDER BY Player_name;
+--------------+---------------------+
| Player_name  | Email               |
+--------------+---------------------+
| Lionel       | lionel10@gmail.com  |
| Ronal        | Ronal7@gmail.com    |
| Sandesh      | sandesh12@gmail.com |
| Sandy        | giri32@gmail.com    |
| Steve Rogers | rogers@gmail.com    |
| Suraz        | suraz332@gmail.com  |
| Wade Wilson  | wilson20@gmail.com  |
+--------------+---------------------+
7 rows in set (0.03 sec)
```

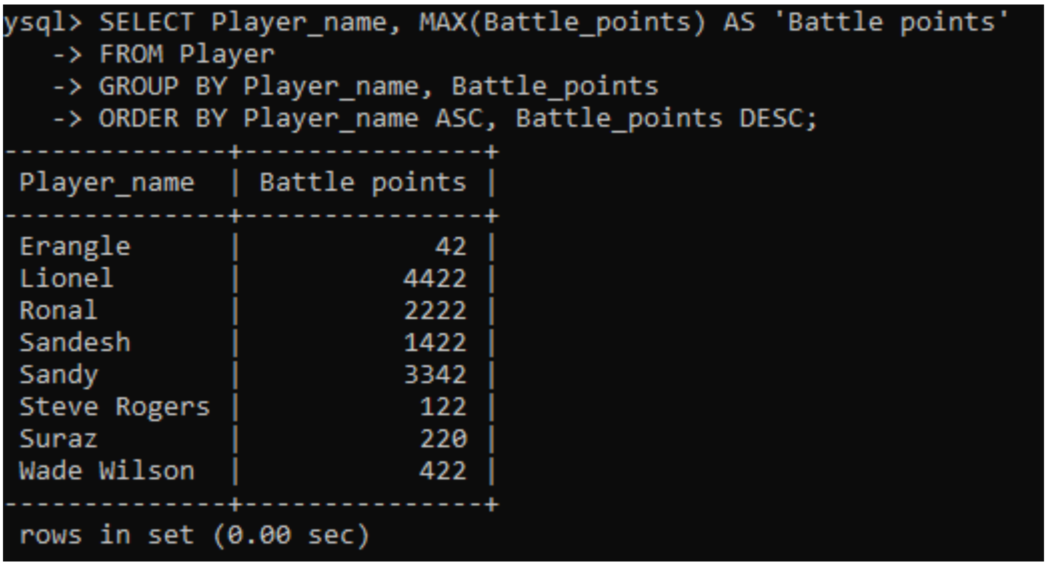
Question 8:



```
mysql> SELECT Hero.Hero_ID,Hero.Hero_name,COUNT(Skin.Skin_ID) AS 'Total number of skin'
    -> FROM Hero INNER JOIN Skin ON Hero.Hero_ID=Skin.Hero_ID
    -> GROUP BY Hero.Hero_ID, Hero.Hero_name;
+---------+-----------+----------------------+
| Hero_ID | Hero_name | Total number of skin |
+---------+-----------+----------------------+
| H001    | Rafaela   |                    1 |
| H002    | Fanny     |                    2 |
| H003    | Dark Rose |                    1 |
| H004    | Tigreal   |                    1 |
| H005    | Estes     |                    1 |
| H006    | Cyclope   |                    1 |
| H007    | Lolita    |                    2 |
| H008    | Clint     |                    1 |
+---------+-----------+----------------------+
8 rows in set (0.78 sec)
```

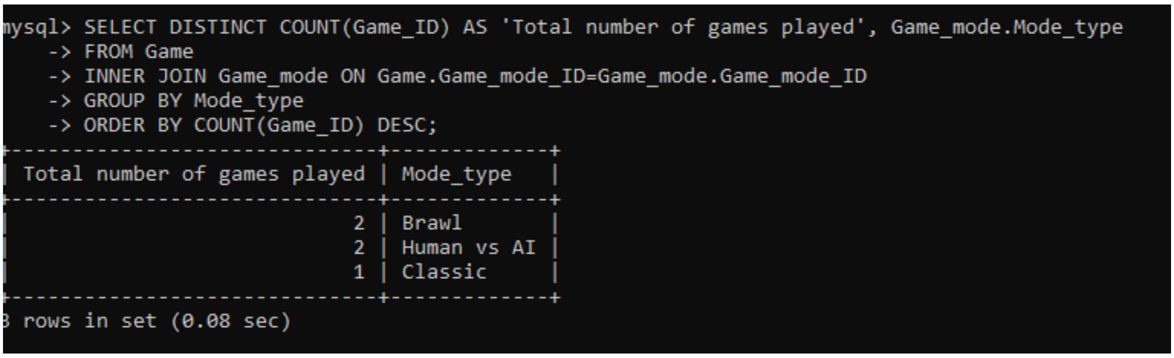
Question 9:

Question 10:



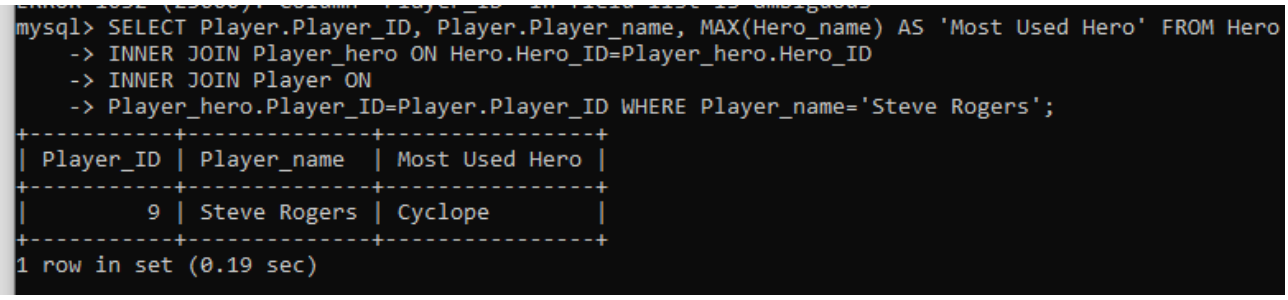
```
ysql> SELECT Player_name, MAX(Battle_points) AS 'Battle points'
   -> FROM Player
   -> GROUP BY Player_name, Battle_points
   -> ORDER BY Player_name ASC, Battle_points DESC;
-------------+---------------+
 Player_name  | Battle points |
-------------+---------------+
 Erangle      |            42 |
 Lionel       |          4422 |
 Ronal        |          2222 |
 Sandesh      |          1422 |
 Sandy        |          3342 |
 Steve Rogers |           122 |
 Suraz        |           220 |
 Wade Wilson  |           422 |
-------------+---------------+
 rows in set (0.00 sec)
```

Question 11:



```
nysql> SELECT DISTINCT COUNT(Game_ID) AS 'Total number of games played', Game_mode.Mode_type
    -> FROM Game
    -> INNER JOIN Game_mode ON Game.Game_mode_ID=Game_mode.Game_mode_ID
    -> GROUP BY Mode_type
    -> ORDER BY COUNT(Game_ID) DESC;
+------------------------------+-------------+
| Total number of games played | Mode_type   |
+------------------------------+-------------+
|                            2 | Brawl       |
|                            2 | Human vs AI |
|                            1 | Classic     |
+------------------------------+-------------+
3 rows in set (0.08 sec)
```

Question 12:



```
ERROR 1052 (23000): Column 'Player_ID' in field list is ambiguous
mysql> SELECT Player.Player_ID, Player.Player_name, MAX(Hero_name) AS 'Most Used Hero' FROM Hero
    -> INNER JOIN Player_hero ON Hero.Hero_ID=Player_hero.Hero_ID
    -> INNER JOIN Player ON
    -> Player_hero.Player_ID=Player.Player_ID WHERE Player_name='Steve Rogers';
+-----------+--------------+----------------+
| Player_ID | Player_name  | Most Used Hero |
+-----------+--------------+----------------+
|         9 | Steve Rogers | Cyclope        |
+-----------+--------------+----------------+
1 row in set (0.19 sec)
```

Workload Matrix

ASSIGNEMNT TOPICS	SURAJ PANDEY	SANDESH SUBEDI	NABIN CHHETRI	SANDESHGIRI
ERD to Relational schema	100%	-	-	-
Normalization of Relations	-	100%	-	-
Database diagram	-	-	-	100%
Data dictionary	-	-	100%	-
Create tables	25%	25%	25%	25%
Data insertion	25%	25%	25%	25%
Queries	25%	25%	25%	25%
Signature	Suraj	Sandesh	Nabin	Sandesh

References

Anon, (n.d.). *Advanced Data Modeling | Relational DBMS with Network Model beneath*. [online] Available at: https://raima.com/advanced-data-modeling/. [Accessed 10 February 2021].

app.diagrams.net. (n.d.). *Flowchart Maker & Online Diagram Software*. [online] Available at: https://app.diagrams.net. [Accessed 10 February 2021].

beginnersbook.com. (2021). Advantages of DBMS over file system. [ONLINE] Available at: https://beginnersbook.com/2015/04/dbms-vs-file-system/. [Accessed 12 February 2021]

Castro, K. (2018). *Advantages of Database Management System*. [online] Tutorialspoint.com. [ONLINE] Available at: https://www.tutorialspoint.com/Advantages-of-Database-Management-System. [Accessed 13 February 2021].

Guru99.com. (2021). *What is DBMS? Application, Types, Example, Advantages*. [online] Available at: <https://www.guru99.com/what-is-dbms.html> [Accessed 14 February 2021].

patil, k., (2021). *ER Diagram to Relational Model Conversion*. [online] DEV Community. Available at: <https://dev.to/ketan_patil/er-diagram-to-relational-model-conversion-49ip> [Accessed 10 February 2021].

Thiru. 2021. What is DBMS? Advantages and Disadvantages of Database Management System(DBMS). [ONLINE] Available at: http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/462-advantages-and-disadvantages-of-dbms.html. [Accessed 14 February 2021].

www.guru99.com. (n.d.). *What is Normalization? 1NF, 2NF, 3NF, BCNF Database Example*. [online] Available at: https://www.guru99.com/database-normalization.html?fbclid=IwAR0iiwdeXOkg98