

# CPSC 304 Milestone 2

Milestone #: 2

Date: February 25, 2024

Group Number: 54

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Dennis Truong	74045766	v5l3b	truong.dennis00@gmail.com
Eric Lew Kim Ping	19138791	n8y4b	ericlew35@gmail.com
Matthew Lai	10958585	v2x6g	mklai10@hotmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

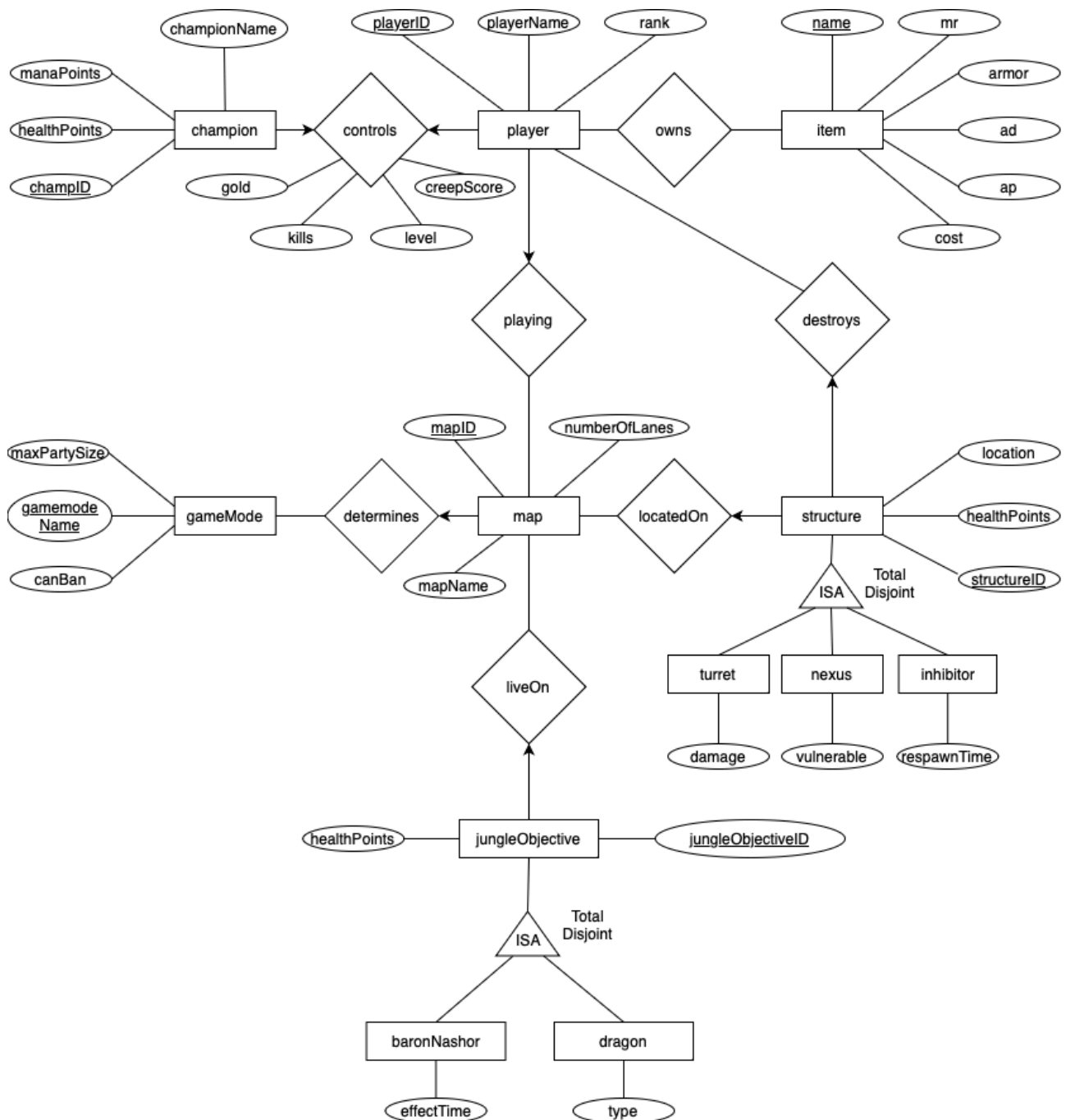
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

# Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>Project Description.....</b>	<b>3</b>
<b>Project Schema.....</b>	<b>5</b>
<b>Functional Dependencies.....</b>	<b>6</b>
<b>Normalization.....</b>	<b>7</b>
<b>All tables post-Normalization.....</b>	<b>9</b>
<b>SQL DDL CREATE Statements.....</b>	<b>10</b>
<b>SQL DDL INSERT Statements.....</b>	<b>13</b>

## Project Description

We aim to create a database application that displays match information for League Of Legends, an online video game where two teams of players fight to destroy the other team's home base. Users may query information about the live state of an online match such as player strength statistics and the progress of destroyed structures.



**Figure 1.** ER Diagram for our proposed League of Legends data scheme

Changes made to ER diagram from Milestone 1:

1. Added a "kill" attribute to the controls relationship.
2. Changed the cardinality of the "owns" relationship from one-to-many to many-to-many because we want players to be able to have many items.
3. Changed the relationship "playing" between Player and GameMode to be a relationship between Player and Map to allow for more flexibility
4. Added "mapID" attribute and made it the primary key
5. Swapped the cardinality of the relationship "determines"
6. Removed the ISA relationship for the "gameMode" entity as it was redundant.
7. Added the vulnerable attribute to the nexus subclass to make the structures more differentiable.
8. Adjusted the naming of the entities to make them more consistent.
9. Added ISA constraints

## Project Schema

Legend	
<u>Underlined</u> : Primary Key,	<b>Bolded</b> : Foreign key

### Note

1. We merged some entities and relationships to reduce redundancy
2. We decided to create a schema for only the subclasses in the ISA's relationships

#### ***Merged player, champion, playing and controls***

playerChampionPlaying(playerID: integer, champID: integer UNIQUE, playerName: char[20], rank: char[20], gold: integer, level: integer, creepScore: integer, kills: integer, championName: char[20], manaPoints: integer, healthPoints: integer, **mapID**: integer)  
- Candidate Key: both playerID or champID alone are candidate keys

#### ***Merged player, items and owns***

ownsItem(name: char[20], playerID: integer, mr: integer, armor: integer, ap: integer, ad: integer, cost : integer)

gameMode(gamemodeName: char[20], maxPartySize: integer, canBan: integer)

#### ***Merged map and determines***

mapDetermines(mapID: integer, mapName: char[20] numberOfLanes: integer, **gamemodeName**: char[20])

#### ***Merged turret, locatedOn and destroyed***

turretStructureLocationDestroyedBy(structureID: integer, healthPoints: integer, location: char[20], damage: integer, **mapID**: integer, **playerID**: integer)

#### ***Merged nexus, locatedOn and destroyed***

nexusStructureLocationDestroyedBy(structureID: integer, healthPoints: integer, location: char[20], vulnerable: integer, **mapID**: integer, **playerID**: integer)

#### ***Merged inhibitor, locatedOn and destroyed***

inhibitorStructureLocationDestroyedBy(structureID: integer, healthPoints: integer, location: char[20], respawnTime: integer, **mapID**: integer, **playerID**: integer)

#### ***Merged baronNashor and livedOn***

baronJungleObjectiveLiveOn(jungleObjectiveID: integer, healthPoints: integer, effectTime: integer, **mapID**: integer)

#### ***Merged dragon and livedOn***

dragonJungleObjectiveLiveOn(jungleObjectiveID: integer, healthPoints: integer, type: char[10], **mapID**: integer)

# Functional Dependencies

ownsItem

- name, playerId → mr, ad, ap, armor, cost

playerChampionPlaying

- playerId → champID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, mapID
- champID → playerId, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, mapID
- creepScore, kills → gold, level

gameMode

- gamemodeName → maxPartySize, canBan

mapDetermines

- mapID → numberOfLanes, mapName, gamemodeName

turretStructureLocatedOnDestroyedBy

- structureID → healthPoints, location, damage, mapID, playerId
- location → healthPoints
- location → damage

nexusStructureLocatedOnDestroyedBy

- structureID → healthPoints, location, vulnerable, mapID, playerId

inhibitorStructureLocatedOnDestroyedBy

- structureID → healthPoints, location, respawnTime, mapID, playerId

baronJungleObjectiveLiveOn

- jungleObjectiveID → healthPoints, effectTime, mapID

dragonJungleObjectiveLiveOn

- jungleObjectiveID → healthPoints, type, mapID
- type → healthPoints

# Normalization

## Normalization for the playerChampionPlaying table:

playerChampionPlaying(playerID, champID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, **mapID**)

- playerID → champID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, **mapID**
- champID → playerID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, mapID
- creepScore, kills → gold, level

Key: champID, playerID

1. The original relation playerChampionPlaying is not in BCNF nor 3NF
2. playerID → (champID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, **mapID**), is in BCNF because playerID is the key
3. champID → (playerID, championName, manaPoints, healthPoints, gold, level, creepScore, kills, rank, **mapID**), is in BCNF because champID is a candidate key
4. creepScore, kills → (gold, level), not in BCNF
  - a. Decompose into

playerStats(playerID, champID, championName, manaPoints, healthPoints, creepScore, kills, rank, **mapID**),

Candidate Key: champID

playerEcon(creepScore, kills, gold, level)

## Normalization for the turretStructureLocatedOnDestroyedBy table:

turretStructureLocatedOnDestroyedBy(structureID, healthPoints, location, damage, mapName, playerID)

- structureID → healthPoints, location,damage, **mapID**, **playerID**
- location → healthPoints
- location → damage

1. The original table turretStructureLocatedOnDestroyedBy is not in BCNF nor 3NF
2. structureID → (healthPoints, location,damage, **mapID**, **playerID**) is in BCNF because structureID is the primary key for the table turretStructure
3. location → healthPoints is not in BCNF nor 3NF because location is not a superkey for the table turretStructureLocatedOnDestroyedBy, and healthPoints is not part of a minimal key for turretStructureLocatedOnDestroyedBy
  - a. Decompose into turretStructureLocatedOnDestroyedBy\_1(location, healthPoints) and turretStructureLocatedOnDestroyedBy\_2(location, damage, structureID, **mapID**, **playerID**)
  - b. For turretStructureLocatedOnDestroyedBy\_2, location → damage violates BCNF so decompose it further into turretStructureLocatedOnDestroyedBy\_3(location, damage) and turretStructureLocatedOnDestroyedBy\_4(location, structureID, **mapID**, **playerID**)

4. location  $\rightarrow$  damage is not in BCNF nor 3NF because location is not a superkey for the table turretStructureLocatedOnDestroyedBy, and damage is not part of a minimal key for turretStructureLocatedOnDestroyedBy.
  - a. After our decomposition in the last part, it does not violate BCNF or 3NF because we have the relation turretStructureLocatedOnDestroyedBy\_3(location, damage), so no need to further reduce
5. Result:
  - turretStructureLocatedOnDestroyedBy\_1(location, healthPoints),
  - turretStructureLocatedOnDestroyedBy\_3(location, damage),
  - turretStructureLocatedOnDestroyedBy\_4(structureID, location, **mapID**, **playerID**)

#### Normalization for the dragonJungleObjectiveLiveOn table:

dragonJungleObjectiveLiveOn(jungleObjectiveID, healthPoints, type, **mapID**)

- jungleObjectiveID  $\rightarrow$  healthPoints, type, **mapID**
  - type  $\rightarrow$  healthPoints
1. The original relation dragonJungleObjectiveLiveOn is not in BCNF nor 3NF
  2. jungleObjectiveID  $\rightarrow$  (healthPoints, type, **mapID**) is in BCNF because the jungleObjectiveID is the primary key for the dragonJungleObjective relation
  3. type  $\rightarrow$  healthPoints is not in BCNF nor 3NF because type is not a superkey in the dragonJungleObjectiveLiveOn relation
    - a. Decompose into dragonJungleObjectiveLiveOn\_1(jungleObjectiveID, type, **mapID**) and dragonJungleObjectiveLiveOn\_2(type, healthPoints)
  4. Since both jungleObjectiveID and type is the minimal superkey in their respective relation, we cannot reduce the relation anymore

BCNF form:

dragonJungleObjectiveLiveOn\_1(jungleObjectiveID, type, **mapID**)  
 dragonJungleObjectiveLiveOn\_2(type, healthPoints)

All other tables are in BCNF, so they will not be normalized



## All tables post-Normalization

playerStats(playerID: integer, champID: integer Unique, championName: char[20],  
manaPoints: integer, healthPoints: integer, creepScore: integer,  
kills: integer, rank: char[20], **mapID**: integer),  
playerEcon(**creepScore**: integer, **kills**: integer, gold: integer, level: integer)

ownsItem(**playerID**: integer, name: char[20], mr: integer, ad: integer, ap: integer, armor:  
integer, cost: integer)

gameMode(gamemodeName: char[20], maxPartySize: integer, canBan: integer)

mapDetermines(mapID: integer, numberOfLanes: integer, **gamemodeName**: char[20])

turretStats(location: char[20], healthPoints: integer); Renamed from  
turretStructureLocatedOnDestroyedBy\_1  
turretDamage(location: char[20], damage: integer); Renamed from  
turretStructureLocatedOnDestroyedBy\_3  
turret(structureID: integer, location: char[20], **mapID**: integer, **playerID**: integer); Renamed  
from turretStructureLocatedOnDestroyedBy\_4

nexus(structureID: integer, healthPoints : integer, location : char[20], vulnerable: integer,  
**mapID**: integer, **playerID**: integer); Renamed from nexusStructureLocatedOnDestroyedBy

inhibitor(structureID: integer, healthPoints: integer, location: char[20], respawnTime: integer,  
**mapID**: integer, **playerID**: integer) ; Renamed from inhibitorStructureLocationDestroyedBy

baronJungleObjective(jungleObjectiveID: integer, healthPoints: integer, effectTime: integer,  
**mapID**: integer)

dragonJungle(jungleObjectiveID: integer, type: char[20], **mapID**: integer); Renamed from  
dragonJungleObjectiveLiveOn\_1  
dragonType(**type**: char[20], healthpoints: integer); Renamed from  
dragonJungleObjectiveLiveOn\_2

## SQL DDL CREATE Statements

```
CREATE TABLE playerStats (  
    playerId INTEGER PRIMARY KEY,  
    champID INTEGER UNIQUE,  
    championName VARCHAR(20),  
    manaPoints INTEGER,  
    healthPoints INTEGER,  
    creepScore INTEGER,  
    kills INTEGER,  
    rank VARCHAR(20),  
    mapID INTEGER,  
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE playerEcon (  
    creepScore INTEGER,  
    kills INTEGER,  
    gold INTEGER,  
    level INTEGER,  
    PRIMARY KEY (creepScore, kills)  
    FOREIGN KEY (creepScore, kills) REFERENCES playerStats(creepScore, kills) ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE ownsItem (  
    playerId INTEGER,  
    name VARCHAR(20),  
    mr INTEGER,  
    ad INTEGER,  
    ap INTEGER,  
    armor INTEGER,  
    cost INTEGER,  
    PRIMARY KEY (playerID, name),  
    FOREIGN KEY (playerID) REFERENCES playerStats(playerID) ON DELETE CASCADE  
);
```

```
CREATE TABLE gameMode (  
    gamemodeName VARCHAR(20) PRIMARY KEY,  
    maxPartySize INTEGER,  
    canBan INTEGER  
);
```

```
CREATE TABLE mapDetermines (  
    mapID INT PRIMARY KEY,  
    mapName VARCHAR(20),
```

```
    numberOfLanes INTEGER,  
    gamemodeName VARCHAR(20)  
    FOREIGN KEY (gamemodeName) REFERENCES gameMode(gamemodeName) ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE turretStats (  
    location VARCHAR(20) PRIMARY KEY,  
    healthPoints INTEGER,  
    FOREIGN KEY(location) REFERENCES turret(location) ON DELETE CASCADE  
);
```

```
CREATE TABLE turretDamage (  
    location VARCHAR(20) PRIMARY KEY,  
    damage INTEGER,  
    FOREIGN KEY(location) REFERENCES turret(location) ON DELETE CASCADE  
);
```

```
CREATE TABLE turret (  
    structureID INTEGER PRIMARY KEY,  
    location VARCHAR(20),  
    mapID INTEGER,  
    playerID INTEGER,  
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE  
    CASCADE,  
    FOREIGN KEY (playerID) REFERENCES playerStats(playerID) ON DELETE CASCADE  
);
```

```
CREATE TABLE nexus (  
    structureID INTEGER PRIMARY KEY,  
    healthPoints INTEGER,  
    location VARCHAR(20),  
    vulnerable INTEGER,  
    mapID INTEGER,  
    playerID INTEGER,  
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE  
    CASCADE,  
    FOREIGN KEY (playerID) REFERENCES playerStats(playerID) ON DELETE CASCADE  
);
```

```
CREATE TABLE inhibitor (  
    structureID INTEGER PRIMARY KEY,  
    healthPoints INTEGER,  
    location VARCHAR(20),  
    respawnTime INTEGER,  
    mapID INTEGER,  
    playerID INTEGER,
```

```
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE  
CASCADE,  
    FOREIGN KEY (playerID) REFERENCES playerStats(playerID) ON DELETE CASCADE  
);
```

```
CREATE TABLE baronJungleObjective (  
    jungleObjectiveID INTEGER PRIMARY KEY,  
    healthPoints INTEGER,  
    effectTime INTEGER,  
    mapID INTEGER,  
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE CASCADE  
);
```

```
CREATE TABLE dragonJungle (  
    jungleObjectiveID INTEGER PRIMARY KEY,  
    type VARCHAR(20),  
    mapID INTEGER,  
    FOREIGN KEY (mapID) REFERENCES mapDetermines(mapID) ON DELETE CASCADE  
);
```

```
CREATE TABLE dragonType (  
    type VARCHAR(20) PRIMARY KEY,  
    healthPoints INTEGER,  
    FOREIGN KEY(type) REFERENCES dragonJungle(type) ON DELETE CASCADE  
);
```

## SQL DDL INSERT Statements

Table	INSERT statements
playerStats( <u>playerID</u> , champID, championName, manaPoints, healthPoints, creepScore, kills, rank, <b>mapID</b> )	INSERT INTO playerStats (playerID, champID, championName, manaPoints, healthPoints, creepScore, kills, rank, mapID) VALUES (1, 123, 'Ahri', 500, 800, 150, 5, 'Gold', 1,), (2, 234, 'Darius', 300, 1000, 120, 7, 'Silver', 1), (3, 345, 'Ezreal', 600, 750, 180, 8, 'Platinum', 2), (4, 456, 'Jinx', 400, 900, 160, 6, 'Gold', 2), (5, 567, 'Yasuo', 200, 850, 140, 9, 'Diamond', 3);
playerEcon( <b>creep Score</b> , <b>kills</b> , gold, level)	INSERT INTO playerEcon (creepScore, kills, gold, level) VALUES (150, 5, 3200, 12), (120, 7, 4300, 15), (180, 8, 2500, 10), (160, 6, 3800, 13), (140, 9, 5200, 18);
ownsItem( <b>playerID</b> , <b>name</b> , mr, ad, ap, armor, cost)	INSERT INTO ownsItem (playerID, name, mr, ad, ap, armor, cost) VALUES (1, 'Banshee's Veil', 60, 0, 0, 45, 2900), (2, 'Infinity Edge', 0, 70, 0, 0, 3400), (3, 'Rabadon's Deathcap', 0, 0, 120, 0, 3800), (4, 'Sunfire Cape', 0, 0, 0, 60, 2900), (5, 'Zhonya's Hourglass', 45, 0, 75, 0, 2600);
gamemode( <u>game modeName</u> , maxPartySize, canBan)	INSERT INTO gameMode (gamemodeName, maxPartySize, canBan) VALUES ('Ranked Solo/Duo', 2, 1), ('Normal Draft', 5, 1), ('ARAM', 5, 0), ('Flex Queue 5v5', 5, 1), ('One for All', 5, 0);
mapDetermines( <u>mapID</u> , numberOfLanes, <b>gamemodeName</b> )	INSERT INTO mapDetermines (mapID, mapName, numberOfLanes, gamemodeName) VALUES (1, 'Summoner's Rift', 3, 'Ranked'), (2, 'Howling Abyss', 3, 'Normal'), (3, 'Twisted Treeline', 1, 'ARAM'), (4, 'Nexus Blitz', 2, 'Blitz'), (5, 'The Crystal Scar', 3, 'Dominion');

turretStats( <b><u>location</u></b> , healthPoints)	INSERT INTO turretStats (location, healthPoints) VALUES ( 'Top Lane', 5000), ( 'Mid Lane', 4500), ( 'Bot Lane', 4000), ( 'Jungle', 4800), ( 'Base', 5200);
turretDamage( <b><u>location</u></b> , damage)	INSERT INTO turretDamage (location, damage) VALUES ( 'Top Lane', 200), ( 'Mid Lane', 180), ( 'Bot Lane', 220), ( 'Jungle', 240), ( 'Base', 210);
turret( <b><u>structureID</u></b> , location, <b>mapID</b> , <b>playerID</b> )	INSERT INTO turret (structureID, location, mapID, playerID) VALUES (1, 'Top Lane Outer', 1, 1), (2, 'Mid Lane Outer', 2, 2), (3, 'Bot Lane Inner', 2, 3), (4, 'Base Bot ', 1, 4), (5, 'Base Nexus Bot', 4, 5);
nexus( <b><u>structureID</u></b> , healthPoints, location, vulnerable, <b>mapID</b> , <b>playerID</b> )	INSERT INTO nexus (structureID, healthPoints, location, vulnerable, mapID, playerID) VALUES (1, 10000, 'Blue Base', 1, 1, 1), (2, 10000, 'Red Base', 1, 1, 4), (1, 10000, 'Blue Base', 1, 2, 2), (2, 10000, 'Red Base', 1, 2, 3), (1, 10000, 'Blue Base', 1, 4, 5);
inhibitor( <b><u>structureID</u></b> , healthPoints, location, respawnTime, <b>mapID</b> , <b>playerID</b> )	INSERT INTO inhibitor (structureID, healthPoints, location, respawnTime, mapID, playerID) VALUES (1, 5000, 'Blue Top', 300, 1, 1), (2, 5000, 'Blue Mid', 300, 1, 1), (3, 5000, 'Blue Bot', 300, 1, 1), (4, 5000, 'Red Top', 300, 1, 3), (5, 5000, 'Red Mid', 300, 1, 3), (6, 5000, 'Red Bot', 300, 1, 3);

baronJungleObjective( <u>jungleObjectiveID</u> , healthPoints, effectTime, <b>mapID</b> )	INSERT INTO baronJungleObjective (jungleObjectiveID, healthPoints, effectTime, mapID) VALUES (1, 6400, 180, 1), (1, 6400, 180, 1), (1, 6400, 180, 1), (1, 6400, 180, 1), (1, 6400, 180, 1);
dragonJungle( <u>jungleObjectiveID</u> , type, <b>mapID</b> )	INSERT INTO dragonJungle (jungleObjectiveID, type, mapID) VALUES (1, 'Infernal', 1), (2, 'Ocean', 1), (3, 'Mountain', 1), (4, 'Cloud', 1), (5, 'Elder', 1);
dragonType( <u>type</u> , healthPoints)	INSERT INTO dragonType (type, healthPoints) VALUES ('Infernal', 4090), ('Ocean', 5730), ('Mountain', 5730), ('Cloud', 5730), ('Elder', 6400);