A DATABASE OF CLASH OF CLANS

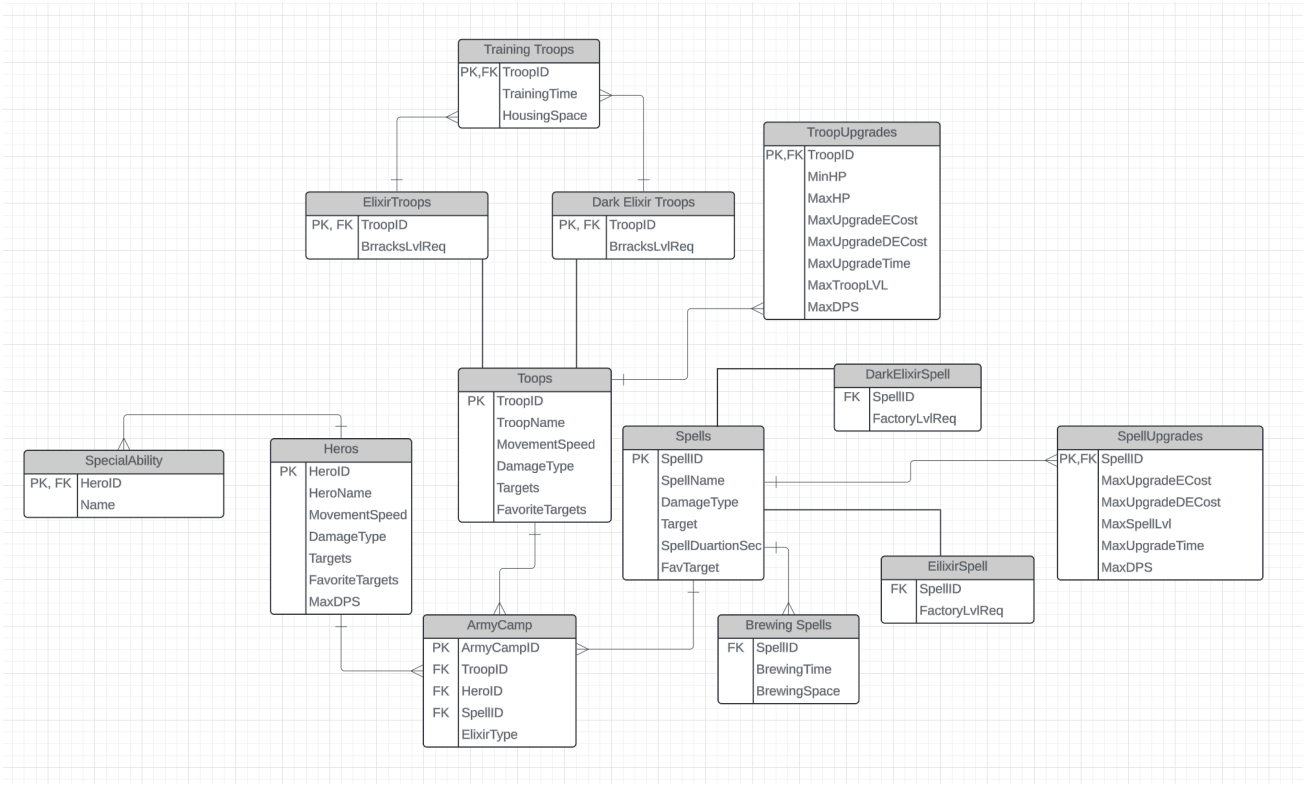
DESIGNED BY DAVID MATUTE JIMENEZ

TABLE OF CONTENTS

Executive Summary

This database, centered around the mobile game Clash of Clans, serves as a comprehensive resource for players. It encompasses data on troops, spells, heroes, and army camps, aiding players in building strong armies. Through strategic querying, players can extract valuable insights into troop attributes, hero abilities, and spell effects, empowering them to assemble the most effective combinations for success in battles. With this information, this database becomes an invaluable tool for guiding players' decision-making processes and enhancing their gameplay experiences.

ENTITY RELATIONSHIP DIAGRAM



TABLE

TROOP TABLE:

Table containing list of troops and common attributes among them


```sql
-- Troops --
create table Troops (
    TroopID         int not null,
    TroopName       text,
    MovementSpeed   int,
    DamageType      text,
    Targets         text,
    FavoriteTargets text,
    primary key(TroopID)
);
```

Data Output | Messages | Notifications

	troopid [PK] integer	troopname text	movementspeed integer	damagetype text	targets text	favoritetargets text
1	1	Barbarian	16	Single Target	Ground	None
2	2	Archer	24	Single Target	Air & Ground	None
3	3	Giant	12	Single Target	Ground	Defense
4	4	Wizard	16	Splash Dama...	Ground & Air	None
5	5	Goblin	32	Single Target	Resources	None
6	6	Balloon	10	Splash Dama...	Ground & Air	Defense
7	7	Dragon	12	Splash Dama...	Ground & Air	None
8	8	P.E.K.K.A	16	Single Target	Ground	Defense
9	9	Minion	32	Single Target	Air	None
10	10	Hog Rider	24	Single Target	Ground	Defense
11	11	Valkyrie	24	Splash Dama...	Ground	None
12	12	Golem	12	Single Target	Ground	Defense
13	13	Bowler	14	Area Splash	Ground	None
14	14	Ice Golem	15	Single Target	Ground	Defenses
15	15	Witch	12	Area Splash	Ground & Air	None

FUNCTIONAL DEPENDENCIES

TroopID → TroopName, MovementSpeed, DamageType, Targets, FavoriteTarget

HERO'S TABLE:

Table containing list of troops and common attributes among them

Query | Query History

```
12  -- Heros --
13  create table Heros (
14      HeroID          int not null,
15      HeroName        text,
16      MovementSpeed   int,
17      DamageType      text,
18      Targets         text,
19      FavoriteTargets text,
20      MaxDPS          int,
21      primary key(HeroID)
22  );
```

Data Output | Messages | Notifications

	heroid [PK] integer	heroname text	movementspeed integer	damagetype text	targets text	favoritetargets text	maxdps integer
1	1	Barbarian King	16	Single Target	Ground	None	608
2	2	Archer Queen	24	Single Target	Ground & Air	None	765
3	3	Grand Warden	16	Single Target	Ground & Air	None	309
4	4	Royal Champion	24	Splash Damage	Ground & Air	Defenses	545

FUNCTIONAL DEPENDENCIES

TroopID → HeroName, MovementSpeed, DamageType, Targets, FavoriteTarget. MaxDPS

SPELLS TABLE:

Table containing list of Spells and common attributes among them


```
25  --spells--
26  create table Spells (
27      SpellID          int not null,
28      SpellName        text,
29      DamageType       text,
30      Target           text,
31      SpellDurationSec int,
32      FavTarget        text,
33      primary key(SpellID)
34  );
```

Data Output | Messages | Notifications

	spellid [PK] integer	spellname text	damagetype text	target text	spellduration sec integer	favtarget text
1	1	Lightning Spell	Splash Damage	Ground & Air	0	None
2	2	Healing Spell	Heal	Ground & Air	10	Troops & Heros
3	3	Rage Spell	Buff	Ground & Air	30	Troops & Heros
4	4	Freeze Spell	Disable	Ground & Air	4	None
5	5	Jump Spell	Movement	Ground	30	Walls
6	6	Clone Spell	Summon	Ground & Air	20	Troops
7	7	Invisibility Spell	Buff	Ground & Air	7	Troops & Heros
8	8	Recall Spell	Redeployment	Ground & Air	0	None
9	9	Poison Spell	Damage over Time	Ground	10	Troops & Heros
10	10	Earthquake	Area Splash	Ground	0	Walls
11	11	Haste Spell	Area Splash	Ground & Air	30	none
12	12	Skeleton Spell	Area Splash	Ground	0	None
13	13	Bat Spell	Area Splash	Air	0	Defenses
14	14	Overgrowth Spell	Area Splash	Ground & Air	28	None

FUNCTIONAL DEPENDENCIES

SpellID → SpellName, DamageType, Targets, FavoriteTarget

ELIXIR TROOP TABLE:

This table stores Elixir Troops that that are unlocked at a certain barrack level

```
36  --ElixerTroops--
37  create table ElixirTroops (
38      TroopID         int not null references Troops(TroopID),
39      BrracksLvlReq   text,
40      primary key(TroopID)
41  );
```

FUNCTIONAL DEPENDENCIES

TroopID → BrracksLvlReq

Data Output | Messages | Notifications

	troopid [PK] integer	brrackslvlreq text
1	1	1
2	2	2
3	3	3
4	4	7
5	5	4
6	6	6
7	7	9
8	8	10

DARK ELIXIR TROOP TABLE:

This table stores Dark Troops that that are unlocked at a certain barrack level

```
43  --DarkElixirTroops--
44  create table DarkElixirTroops (
45      TroopID         int not null references Troops(TroopID),
46      BrracksLvlReq   text,
47      primary key(TroopID)
48  );
```

Data Output | Messages | Notifications

	troopid [PK] integer	brrackslvlreq text
1	9	1
2	10	2
3	11	3
4	12	4
5	13	7
6	14	8
7	15	5

FUNCTIONAL DEPENDENCIES

TroopID → BrracksLvlReq

ELIXIR SPELL TABLE:

This table stores Elixir spells that that get unlocked at a certain spell factory level

```
65  --ElixirSpell--
66  create table ElixirSpell (
67      SpellID         int not null references Spells(SpellID),
68      FactoryLvlReq   text,
69      primary key(SpellID)
70  );
```

Data Output | Messages | Notifications

	spellid [PK] integer	factorylvlreq text
1	1	1
2	2	2
3	3	3
4	4	4
5	5	4
6	6	5
7	7	6
8	8	7

FUNCTIONAL DEPENDENCIES

SpellID → FactoryLvlReq

DARK ELIXIR SPELL TABLE:

This table stores Dark Elixir spells that that get unlocked at a certain spell factory level

```
72  --DarkElixirSpell--
73  create table DarkElixirSpell (
74      SpellID          int not null references Spells(SpellID),
75      FactoryLvlReq    text,
76      primary key(SpellID)
77  );
```

Data Output | Messages | Notifications

	spellid [PK] integer	factorylvlreq text
1	9	1
2	10	2
3	11	3
4	12	4
5	13	5
6	14	6

FUNCTIONAL DEPENDENCIES

SpellID → FactoryLvlReq

SPECIAL ABILITY TABLE:

This table stores HeroID and the special ability they have

```
58  --SpecialAbility--
59  create table SpecialAbility (
60      HeroID          int not null references Heros(HeroID)
61      Name            text,
62      primary key(HeroID)
63  );
```

Data Output | Messages | Notifications

	heroid [PK] integer	name text
1	1	Iron Fist
2	2	Royal Cloak
3	3	Eternal Tome
4	4	Royal Cloak

FUNCTIONAL DEPENDENCIES

HeroID → Name

TROOP UPGRADE TABLE:

This table stores the troops upgrades and certain stats

```
99  --TroopUpgrades--
100 create table TroopUpgrades (
101     TroopID             int not null references Troops(TroopID),
102     MinHP               int,
103     MaxHP               int,
104     MaxUpgradeECost     int,
105     MaxUpgradeDECost    int,
106     MaxUpgradeTime      int,
107     MaxTroopLvl         int,
108     MaxDPS              int,
109     primary key(TroopID)
110 );
```

Data Output | Messages | Notifications

	troopid [PK] integer	minhp integer	maxhp integer	maxupgradeecost integer	maxupgradedecost integer	maxupgradetime integer	maxtrooplvl integer	maxdps integer
1	1	45	290	18000000	0	13	12	48
2	2	20	68	18000000	0	13	12	40
3	3	300	2400	18500000	0	14	12	94
4	4	75	290	19200000	0	14	12	275
5	5	22	146	16000000	0	14	9	72
6	6	150	1140	20000000	0	16	11	290
7	7	1900	5300	21500000	0	16	11	390
8	8	3000	7700	20500000	0	14	11	810
9	9	58	120	0	320000	14	12	78
10	10	270	1380	0	350000	15	13	213
11	11	750	2800	0	340000	15	11	238
12	12	5100	9200	0	350000	15	13	95
13	13	325	600	0	350000	15	8	126
14	14	2600	4200	0	360000	16	8	200
15	15	300	560	0	360000	16	7	220

FUNCTIONAL DEPENDENCIES

TroopID → MinHP, MaxHP, MaxUpgradeECost,MaxUpgradeDECost, MaxUpgradeTime, MaxTroopLvl, MaxDPS

SPELL UPGRADE TABLE:

This table stores the spells upgrades and certain stats

```
88  --SpellUpgrades--
89  create table SpellUpgrades (
90      SpellID             int not null references Spells(SpellID),
91      MaxUpgradeECost     int,
92      MaxUpgradeDECost    int,
93      MaxSpellLvl         int,
94      MaxUpgradeTime      int,
95      MaxDPS              int,
96      primary key(SpellID)
97  );
```

Data Output | Messages | Notifications

	spellid [PK] integer	maxupgradeecost integer	maxupgradedecost integer	maxspelllvl integer	maxupgradetime integer	maxdps integer
1	1	900000	0	8	11	150
2	2	1000000	0	8	11	0
3	3	1200000	0	8	11	0
4	4	1300000	0	8	10	0
5	5	1400000	0	8	12	0
6	6	1500000	0	8	12	0
7	7	1600000	0	8	13	0
8	8	1700000	0	8	10	0
9	9	0	200000	8	9	200
10	10	0	220000	7	12	400
11	11	0	280000	7	8	0
12	12	0	260000	7	8	0
13	13	0	260000	7	11	0
14	14	0	300000	4	10	0

FUNCTIONAL DEPENDENCIES

SpellID → MaxUpgradeECost,MaxUpgradeDECost, MaxUpgradeTime, MaxSpellLvl, MaxDPS

TRAINING TROOPS TABLE:

This table stores the troops time it takes to train and how much space it takes up

```
50  --TrainingTroops--
51  create table TrainingTroops (
52      TroopID         int not null references Troops(TroopID),
53      TrainingTimeSec int,
54      HousingSpace    int,
55      primary key(TroopID)
56  );
```

Data Output | Messages | Notifications

	troopid [PK] integer	trainingtimesec integer	housingspace integer
1	1	5	1
2	2	6	1
3	3	30	5
4	4	30	5
5	5	7	1
6	6	30	5
7	7	180	20
8	8	180	25
9	9	18	2
10	10	45	5
11	11	90	8
12	12	300	30
13	13	60	6
14	14	150	15
15	15	120	12

FUNCTIONAL DEPENDENCIES

TroopID → TrainingTimeSec, HousingSpace

BREWING SPELLS TABLE:

This table stores the spells time it takes to brew and how much space it takes up

```
79  -- Brewing Spells --
80  create table BrewingSpells (
81      SpellID          int not null references Spells(SpellID),
82      BrewingTimeSec   int,
83      BrewingSpace     int,
84      primary key(SpellID)
85
86  );
```

FUNCTIONAL DEPENDENCIES

SpellID → BrewingTimeSec, BrewingSpace

Data Output | Messages | Notifications

	spellid [PK] integer	brewingtimesec integer	brewingspace integer
1	1	180	1
2	2	240	2
3	3	3000	2
4	4	180	1
5	5	240	2
6	6	360	3
7	7	360	1
8	8	360	2
9	9	120	1
10	10	120	1
11	11	100	1
12	12	120	1
13	13	120	1
14	14	300	2

ARMY CAMP TABLE:

This table stores the troops, heros, and spells to make an army

```
112 --ArmyCamp--
113 create table ArmyCamp(
114     ArmyCampID      int not null,
115     TroopID         int not null references Troops(TroopID),
116     HeroID          int not null references Heros(HeroID),
117     SpellID         int not null references Spells(SpellID),
118     ElixirType      text,
119     primary key(ArmyCampID)
120 );
```

Data Output | Messages | Notifications

	armycampid [PK] integer	troopid integer	heroid integer	spellid integer	elixirtype text
1	1	1	1	1	Elixir & Dark Elixir
2	2	2	2	2	Elixir & Dark Elixir
3	3	3	3	3	Elixir
4	4	4	4	4	Elixir & Dark Elixir
5	5	5	1	5	Elixir & Dark Elixir
6	6	6	3	6	Elixir

FUNCTIONAL DEPENDENCIES

ArmyCampID → TroopID, HeroID, SpellID, ElixirType

VIEWS

ArmyComposition: Shows a viewing of the troops, heroes, and spells that is assigned to each army camp

```
305 --views--
306 create view ArmyComposition as
307 select ac.ArmyCampID, t.TroopName, h.HeroName, s.SpellName, ac.ElixirType
308 from ArmyCamp ac
309 join Troops t on ac.TroopID = t.TroopID
310 join Heros h on ac.HeroID = h.HeroID
311 join Spells s on ac.SpellID = s.SpellID;
312
313 select *
314 from ArmyComposition;
```

Data Output | Messages | Notifications

	armycampid integer	troopname text	heroname text	spellname text	elixirtype text
1	1	Barbarian	Barbarian King	Lightning Spell	Elixir & Dark Elixir
2	2	Archer	Archer Queen	Healing Spell	Elixir & Dark Elixir
3	3	Giant	Grand Warden	Rage Spell	Elixir
4	4	Wizard	Royal Champi...	Freeze Spell	Elixir & Dark Elixir
5	5	Goblin	Barbarian King	Jump Spell	Elixir & Dark Elixir
6	6	Balloon	Grand Warden	Clone Spell	Elixir

HeroDetails: This view combines heros table with special Ability table to provide details on each hero


```
318 create view HeroDetails as
319 select Heros.*, SpecialAbility.Name as SpecialAbilityName
320 from Heros
321 inner join SpecialAbility on Heros.HeroID = SpecialAbility.HeroID;
322
323 select *
324 from HeroDetails;
```

Data Output | Messages | Notifications

	heroid integer	heroname text	movementspeed integer	damagetype text	targets text	favoritetargets text	maxdps integer	specialabilityname text
1	1	Barbarian King	16	Single Target	Ground	None	608	Iron Fist
2	2	Archer Queen	24	Single Target	Ground & Air	None	765	Royal Cloak
3	3	Grand Warden	16	Single Target	Ground & Air	None	309	Eternal Tome
4	4	Royal Champion	24	Splash Damage	Ground & Air	Defenses	545	Royal Cloak

TrainingAndBrewing: This view combines TrainingTroops and Brewing spells and shows how long it takes to create that spell or troop in ascending order to compare


```sql
--TrainingAndBrewing--
create TrainingAndBrewing as
select 'Training Troop' as type,
       TrainingTroops.TroopID as identification,
       Troops.TroopName as Name,
       TrainingTimeSec as DurationInSeconds
from TrainingTroops
join Troops on TrainingTroops.TroopID = Troops.TroopID
union all
select 'Brewing Spell' as type,
       BrewingSpells.SpellID as identification,
       Spells.SpellName as Name,
       BrewingTimeSec as DurationInSeconds
from BrewingSpells
join Spells on BrewingSpells.SpellID = Spells.SpellID
order by DurationInSeconds asc;
```

Data Output | Messages | Notifications

	type text	identification integer	name text	durationinseconds integer
1	Training Troop	1	Barbarian	5
2	Training Troop	2	Archer	6
3	Training Troop	5	Goblin	7
4	Training Troop	9	Minion	18
5	Training Troop	4	Wizard	30
6	Training Troop	6	Balloon	30
7	Training Troop	3	Giant	30
8	Training Troop	10	Hog Rider	45
9	Training Troop	13	Bowler	60
10	Training Troop	11	Valkyrie	90
11	Brewing Spell	11	Haste Spell	100
12	Training Troop	15	Witch	120
13	Brewing Spell	9	Poison Spell	120
14	Brewing Spell	10	Earthquake	120
15	Brewing Spell	12	Skeleton Spell	120
16	Brewing Spell	13	Bat Spell	120
17	Training Troop	14	Ice Golem	150
18	Training Troop	7	Dragon	180
19	Brewing Spell	4	Freeze Spell	180
20	Brewing Spell	1	Lightning Spell	180
21	Training Troop	8	P.E.K.K.A	180
22	Brewing Spell	2	Healing Spell	240
23	Brewing Spell	5	Jump Spell	240
24	Brewing Spell	14	Overgrowth Sp...	300
25	Training Troop	12	Golem	300
26	Brewing Spell	8	Recall Spell	360
27	Brewing Spell	7	Invisibility Spell	360
28	Brewing Spell	6	Clone Spell	360
29	Brewing Spell	3	Rage Spell	3000

Total rows: 29 of 29 | Query complete 00:00:00.063

SpellDetails: This view combines Spells and spellUpgrades to show the list of details of that spell


```
347 --Spell Details--
348 create SpellDetails as
349 select Spells.SpellID as SpellID, Spells.SpellName, Spells.DamageType, Spells.Target,
350 Spells.SpellDurationSec, Spells.FavTarget,SpellUpgrades.MaxUpgradeECost, SpellUpgrades.MaxUpgradeDECost,
351 SpellUpgrades.MaxSpellLvl, SpellUpgrades.MaxUpgradeTime, SpellUpgrades.MaxDPS
352 from Spells
353 inner join SpellUpgrades on Spells.SpellID = SpellUpgrades.SpellID;
354
355 select *
356 from spellDetails;
```

	id integer	spellname text	damagetype text	target text	spellduration sec integer	favtarget text	maxupgradeecost integer	maxupgradedecost integer	maxspelllvl integer	maxupgradetime integer	maxdps integer
1	1	Lightning Spell	Splash Damage	Ground & Air	0	None	900000	0	8	11	150
2	2	Healing Spell	Heal	Ground & Air	10	Troops & Heros	1000000	0	8	11	0
3	3	Rage Spell	Buff	Ground & Air	30	Troops & Heros	1200000	0	8	11	0
4	4	Freeze Spell	Disable	Ground & Air	4	None	1300000	0	8	10	0
5	5	Jump Spell	Movement	Ground	30	Walls	1400000	0	8	12	0
6	6	Clone Spell	Summon	Ground & Air	20	Troops	1500000	0	8	12	0
7	7	Invisibility Spell	Buff	Ground & Air	7	Troops & Heros	1600000	0	8	13	0
8	8	Recall Spell	Redeployment	Ground & Air	0	None	1700000	0	8	10	0
9	9	Poison Spell	Damage over Time	Ground	10	Troops & Heros	0	200000	8	9	200
10	10	Earthquake	Area Splash	Ground	0	Walls	0	220000	7	12	400
11	11	Haste Spell	Area Splash	Ground & Air	30	none	0	280000	7	8	0
12	12	Skeleton Spell	Area Splash	Ground	0	None	0	260000	7	8	0
13	13	Bat Spell	Area Splash	Air	0	Defenses	0	260000	7	11	0
14	14	Overgrowth Sp…	Area Splash	Ground & Air	28	None	0	300000	4	10	0

REPORTS AND INTERESTING QUERIES

REPORT 1:

Query: Returns an overview of the maximum upgrade cost of each troop to see which troop is worth upgrading

```
358  --Reports--
359  select Troops.TroopName, TroopUpgrades.MaxUpgradeECost, TroopUpgrades.MaxUpgradeDECost
360  from Troops
361  join TroopUpgrades on Troops.TroopID = TroopUpgrades.TroopID;
```

Data Output | Messages | Notifications

	troopname text	maxupgradeecost integer	maxupgradedecost integer
1	Barbarian	18000000	0
2	Archer	18000000	0
3	Giant	18500000	0
4	Wizard	19200000	0
5	Goblin	16000000	0
6	Balloon	20000000	0
7	Dragon	21500000	0
8	P.E.K.K.A	20500000	0
9	Minion	0	320000
10	Hog Rider	0	350000
11	Valkyrie	0	340000
12	Golem	0	350000
13	Bowler	0	350000
14	Ice Golem	0	360000
15	Witch	0	360000

REPORT 2:

Query: Returns all troops with there damage per second from ascending order to see which troop is worth training

```
363  select Troops.TroopName, TroopUpgrades.MaxDPS
364  from Troops
365  join TroopUpgrades on Troops.TroopID = TroopUpgrades.TroopID
366  order by TroopUpgrades.MaxDPS asc;
```

Data Output | Messages | Notifications

	troopname text	maxdps integer
1	Archer	40
2	Barbarian	48
3	Goblin	72
4	Minion	78
5	Giant	94
6	Golem	95
7	Bowler	126
8	Ice Golem	200
9	Hog Rider	213
10	Witch	220
11	Valkyrie	238
12	Wizard	275
13	Balloon	290
14	Dragon	390
15	P.E.K.K.A	810

REPORT 3:

Query: Returns the Hero with the highest dps, who doesn't want the strongest hero in their army?

```
368 select Heros.HeroName, Heros.MaxDPS, SpecialAbility.Name as SpecialAbility
369 from Heros
370 join SpecialAbility on Heros.HeroID = SpecialAbility.HeroID
371 order by Heros.MaxDPS desc
372 limit 1;
```

Data Output | Messages | Notifications

	heroname text	maxdps integer	specialability text
1	Archer Queen	765	Royal Cloak

REPORT 4:

Query: Returns the spell with the longest duration which is the best for attacking

```
374 select Spells.SpellName, Spells.SpellDurationSec, Spells.FavTarget
375 from Spells
376 where Spells.SpellDurationSec = (
377     select max(SpellDurationSec)
378     from Spells
379 )
380 limit 1;
```

Data Output | Messages | Notifications

	spellname text	spellduration sec integer	favtarget text
1	Rage Spell	30	Troops & Heros

STORED PROCEDURE

STORED PROCEDURE 1:

RETURNS INFORMATION ABOUT A SPECIFIC TROOP BASED ON ITS TROOPID

```
--stored procedures--
create or replace function GetTroopInfo(p_TroopID INT)
returns table(
    TroopID int,
    TroopName text,
    MovementSpeed int,
    DamageType text,
    Targets text,
    FavoriteTargets text
) as
$$
begin
    return query
    select Troops.TroopID, Troops.TroopName, Troops.MovementSpeed, Troops.DamageType, Troops.Targets, Troops.FavoriteTargets
    from Troops
    where Troops.TroopID = p_TroopID;
end;
$$
language plpgsql;
```

SAMPLE OUTPUT:

Data Output | Messages | Notifications

	troopid integer	troopname text	movementspeed integer	damagetype text	targets text	favoritetargets text
1	6	Balloon	10	Splash Damage	Ground & Air	Defense

STORED PROCEDURE

STORED PROCEDURE 2:

RETURNS INFORMATION ABOUT A SPECIFIC SPELL BASED ON ITS SPELLID

```
406 --GetSpellInfo--
407 create or replace function GetSpellInfo(p_SpellID int)
408 returns table (
409     SpellID int,
410     SpellName text,
411     DamageType text,
412     Target text,
413     SpellDurationSec int,
414     FavTarget text
415 ) as $$
416 begin
417     return query
418     select Spells.SpellID, Spells.SpellName, Spells.DamageType, Spells.Target, Spells.SpellDurationSec, Spells.FavTarget
419     from Spells
420     where Spells.SpellID = p_SpellID;
421 end;
422 $$ lanaguage plpgsql;
423
424 select *
425 from GetSpellInfo(6);
```

SAMPLE OUTPUT:

Data Output | Messages | Notifications

	spellid integer	spellname text	damagetype text	target text	spelldurationsec integer	favtarget text
1	6	Clone Spell	Summon	Ground & Air	20	Troops

STORED PROCEDURE

STORED PROCEDURE 3:

RETURNS INFORMATION ABOUT A SPECIFIC HERO BASED ON ITS HEROID

```
--GetHeroInfo--
create or replace function GetHeroInfo(p_HeroID int)
returns table (
    HeroID int,
    HeroName text,
    MovementSpeed int,
    DamageType text,
    Targets text,
    FavoriteTargets text,
    MaxDPS int
) as $$
begin
    return query
    select Heros.HeroID, Heros.HeroName, Heros.MovementSpeed, Heros.DamageType, Heros.Targets, Heros.FavoriteTargets, Heros.MaxDPS
    from Heros
    where Heros.HeroID = p_HeroID;
end;
$$ language plpgsql;

select * from GetHeroInfo(2);
```

SAMPLE OUTPUT:

Data Output | Messages | Notifications

	heroid integer	heroname text	movementspeed integer	damagetype text	targets text	favoritetargets text	maxdps integer
1	2	Archer Queen	24	Single Target	Ground & Air	None	765

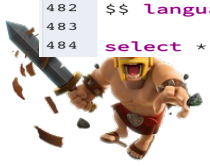
STORED PROCEDURE

STORED PROCEDURE 4:

RETURNS INFORMATION ABOUT A SPECIFIC ARMYCAMP BASED ON THE TROOPID, HEROID, AND SPELLID

```
--GetArmyCampInfo--
create or replace function GetArmyCampInfo(p_ArmyCampID int)
returns table (
    ArmyCampID int,
    TroopID int,
    TroopName text,
    HeroID int,
    HeroName text,
    SpellID int,
    SpellName text,
    ElixirType text
) as $$
begin
    return query
    select
        ArmyCamp.ArmyCampID,
        ArmyCamp.TroopID,
        Troops.TroopName,
        ArmyCamp.HeroID,
        Heros.HeroName,
        ArmyCamp.SpellID,
        Spells.SpellName,
        ArmyCamp.ElixirType
    from
        ArmyCamp
    left join
        Troops on ArmyCamp.TroopID = Troops.TroopID
    left join
        Heros on ArmyCamp.HeroID = Heros.HeroID
    left join
        Spells on ArmyCamp.SpellID = Spells.SpellID
    where
        ArmyCamp.ArmyCampID = p_ArmyCampID;
end;
$$ language plpgsql;

select * from GetArmyCampInfo(3);
```



SAMPLE OUTPUT:

Data Output | Messages | Notifications

	armycampid integer	troopid integer	troopname text	heroid integer	heroname text	spellid integer	spellname text	elixirtype text
1	3	3	Giant	3	Grand Warden	3	Rage Spell	Elixir

TRIGGER

TRIGGER MAKES SURE WHEN PLAYER ADDS A NEW TROOP TO THE DATABASE THAT ONLY POSITIVE INTEGERS CAN BE ADDED FOR MOVEMENT SPEED

```
--TRIGGER--
--ensure MovementSpeed is positive.
create or replace function validate_movement_speed()
returns trigger as
$$
begin
    if new.MovementSpeed <= 0 then
        raise exception 'Movement speed must be positive';
    end if;
    return new;
end;
$$ language plpgsql;

create trigger validate_movement_speed_trigger
before insert or update Troops
for each row execute function validate_movement_speed();

--positive MovementSpeed--
insert into Troops (TroopID, TroopName, MovementSpeed, DamageType, Targets, FavoriteTargets)
values (16, 'New Troop', 10, 'Single Target', 'Ground', 'None');

--non-positive MovementSpeed--
insert into Troops (TroopID, TroopName, MovementSpeed, DamageType, Targets, FavoriteTargets)
values (17, 'Another Troop', -5, 'Single Target', 'Ground', 'None');
```

SAMPLE OUTPUT:

Data Output | Messages | Notifications

```
ERROR:  Movement speed must be positive
CONTEXT:  PL/pgSQL function validate_movement_speed() line 4 at RAISE

SQL state: P0001
```

SECURITY

PLAYER ROLE: REPRESENTS THE DATABASE TO THE PLAYER WHO HAS FULL ACCESS

```
512  --SECURITY--
513  create role player;
514  grant all
515  on all tables in schema public
516  to player;
```

IMPLEMENTATION NOTES

. CLASH OF CLANS IS AN ONLINE MULTIPLAYER STRATEGY GAME

. EVERY TROOP, SPELL, AND HERO IS UNIQUE. UNABLE TO ADD THE SAME ENTITY IN THE TABLE.

.FOR SIMPLICITY ARMY CAMP CAN ONLY HOLD 1 TROOP, 1 HERO, AND 1 SPELL. THE ACTUAL GAME HAS AN ARMY CAMP THAT HAS A HOUSING SPACE OF 320 MEANING MULTIPLE TROOPS AND EVEN THE SAME REPEATING TROOPS CAN BE IN AN ARMY CAMP. THIS SAME APPLIES TO SPELLS. HEROS DO NOT HAVE HOUSING SPACE

KNOWN PROBLEMS

. DUE TO THE GAME BEING SO BIG INCLUDING HAVING MORE THAN 1 ARMY CAMP FOR DIFFERENT BASES, TROOPS, SPELLS, AND HEROS. THERE WAS A RESTRICTION ON THE NUMBER OF TABLES. THERE COULD HAVE BEEN TABLES FOR "TOWNHALLS", "HERO UPGRADES", ARMY CAMP WITH A HOUSING CAPACITY OF 320. ADDING THIS TO THE DATABASE WOULD ALLOW PLAYERS TO HAVE GREATER DETAIL IN CHOOSING TROOPS, HEROS, AND SPELLS FOR THEIR ARMY CAMPS
. PLAYERS CAN HAVE MORE THAN 1 ARMY CAMP WHICH WAS NOT PART OF THE DATABASE

FUTURE ENHACEMENTS

.THERE ARE MORE TROOPS IN THE GAME THAT CAN BE ADDED TO THE DATABASE WHICH WILL ALLOW THE PLAYER TO HAVE MORE OPTIONS IN CHOOSING ON WHAT TO ADD TO THEIR ARMY CAMP.

. THE GAME IS CONSTANTLY GROWING SO THE PLAYER CAN EASILY ADD THE NEW TROOPS, SPELLS, OR TROOPS TO THE DATABASE. THE STATS ARE ALSO ALWAYS CHANGING IN THE GAME

. THERE IS A LOT OF POTENTIAL, WE COULD HAVE 1 PLAYER HAVE MULTIPLE ARMY CAMPS. THEY CAN ALSO HAVE MULTIPLE OF THE SAME TROOPS OR SPELLS IN THE ARMY CAMP.