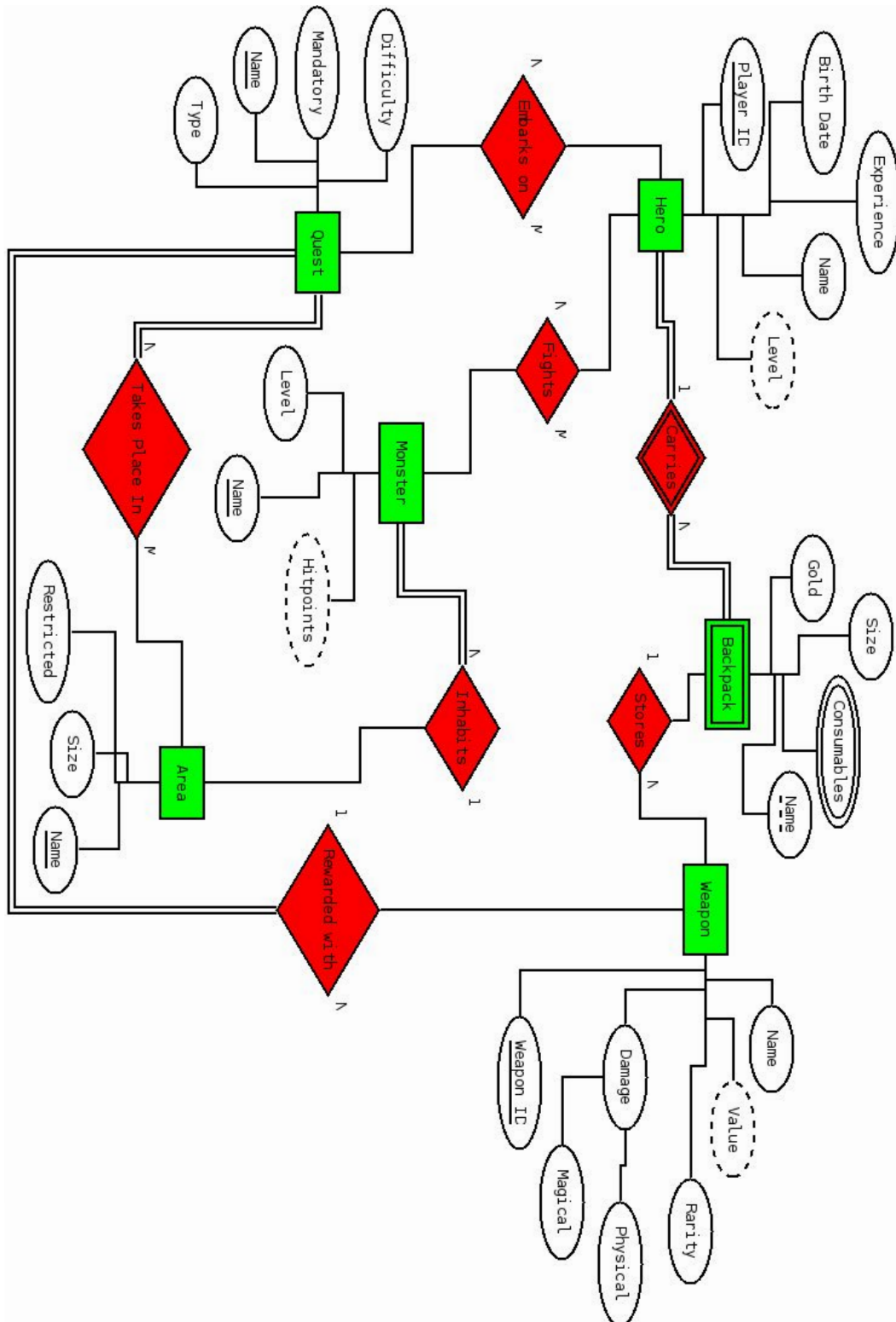


## COSC 344 Assignment 2

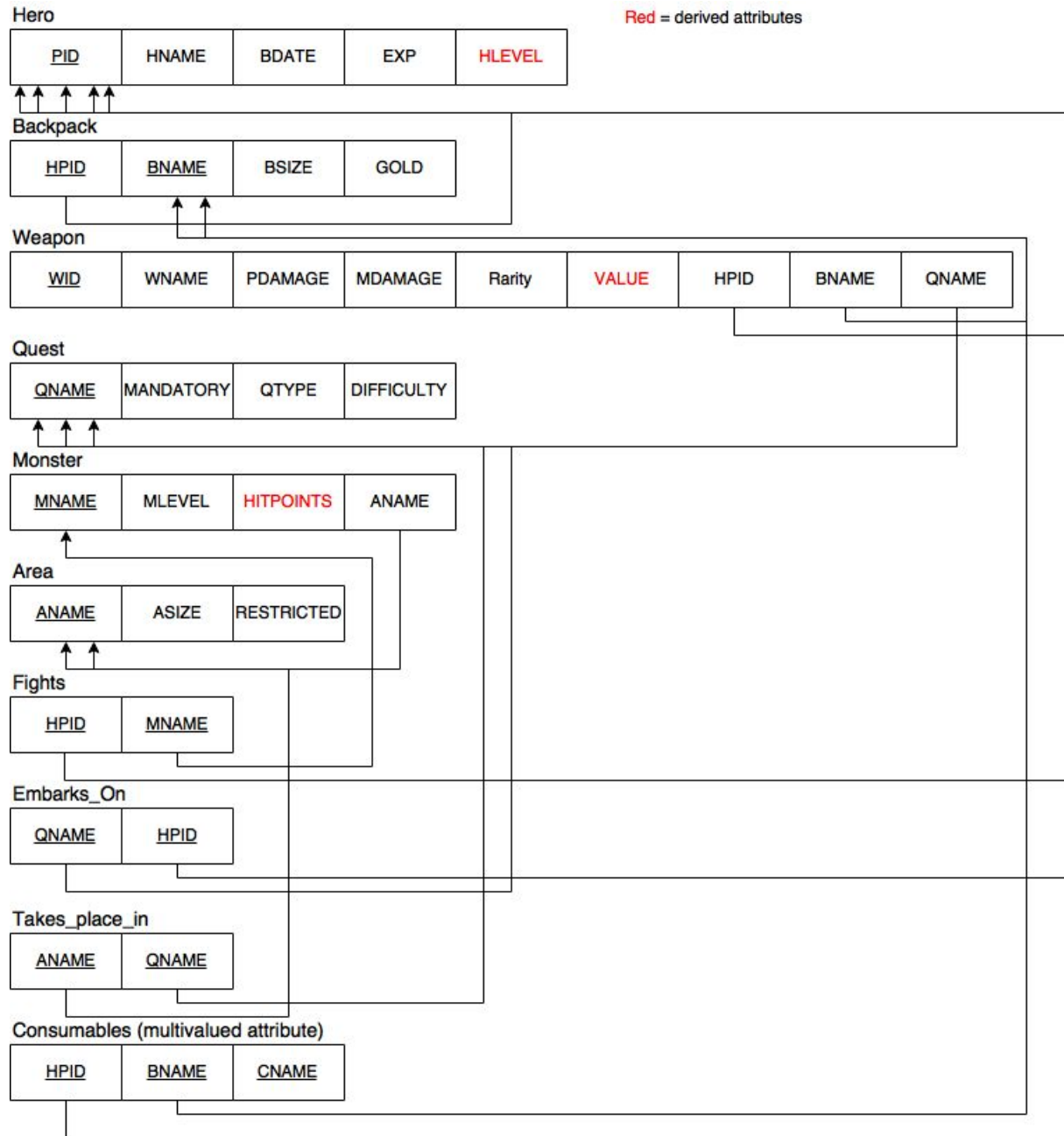
Group Leader: Sonny Lim

Group 9: Robert Young, Amy Corkery, Thomas Scott, Albert Gao, Jialin Yang

## Part 1: ER Diagram



## Part 2: Creation of Relation Schema



# Part 3: Normalisation

## (1) 1st Normal Form

Change the CONSUMABLES **multivalued attribute** to an atomic one by introducing it as a separate relation (Step 6 of the relational schema), so already done in the above relation schema.

There are no **composite attributes** after the relational mapping is done. (physical\_damage and magical\_damage).

There are no **nested relations**.

First normal form is therefore achieved.

## (2) 2nd Normal Form

There are no relations with multiple candidate keys, so no relations where there are attributes only partially dependent on the primary key. All non prime attributes are fully dependent on the primary key.

Thus, 2NF is achieved.

## (3) 3rd Normal Form

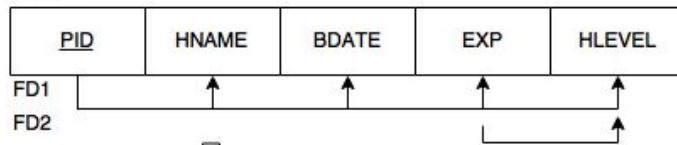
Derived attributes are transitively dependent on the primary key (through the attributes they're derived from), so are moved to their own relations. See the diagram below.

After doing this:

- Every non-prime attribute of each relation R is non-transitively dependent on every key of R.
- That is, for every functional dependency  $X \rightarrow A$  in a relation R, X is a superkey, as for our relation schema, X is always the primary key.

Thus, 3NF is achieved.

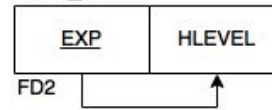
### Hero



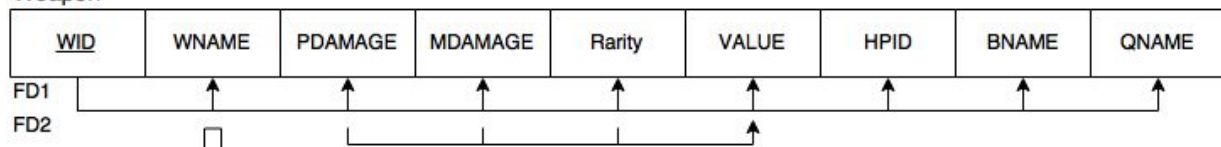
### Hero1



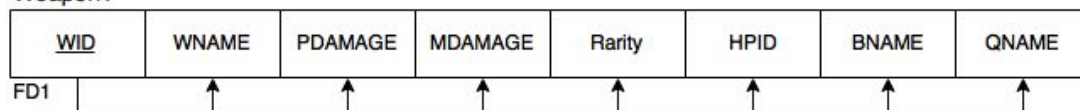
### Hero\_Level



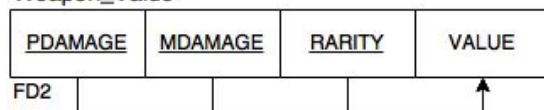
### Weapon



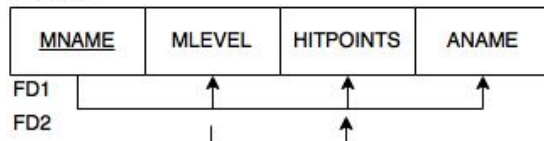
### Weapon1



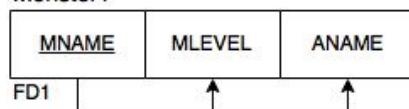
### Weapon\_Value



### Monster



### Monster1



### Monster\_Hitpoints



## (4) BCNF

This is also achieved, as in our 3NF form, we have no functional dependencies  $X \rightarrow A$  in any relation  $R$  such that  $A$  is a prime attribute of  $R$  (the condition that separates 3NF from BCNF). Another way of seeing this is that (from lec 9, s20) every LHS of a functional dependency is a candidate key.

Thus, BCNF is achieved.

## Part 4: SQL

```
DROP TABLE HLEVEL;  
DROP TABLE MHP;  
DROP TABLE WVALUE;  
DROP TABLE consumable;  
DROP TABLE takes_place_in;  
DROP TABLE fights;  
DROP TABLE embarks_on;  
DROP TABLE monster;  
DROP TABLE area;  
DROP TABLE weapon;  
DROP TABLE quest;  
DROP TABLE backpack;  
DROP TABLE hero;
```

```
CREATE TABLE hero  
(pid INT PRIMARY KEY,  
hname VARCHAR2(31) NOT NULL,  
bdate DATE NOT NULL,  
exp INT NOT NULL);
```

```
CREATE TABLE HLEVEL(  
hexp INT PRIMARY KEY,  
hlevel INT NOT NULL  
);
```

```
CREATE TABLE backpack
(bname VARCHAR2(31) NOT NULL,
hpid INT REFERENCES hero(pid),
bsize INT NOT NULL,
gold INT NOT NULL,
PRIMARY KEY(hpid,bname));
```

```
CREATE TABLE quest
(qname VARCHAR2(31) PRIMARY KEY,
mandatory INT NOT NULL,
qtype VARCHAR2(31) NOT NULL,
difficulty VARCHAR2(31) NOT NULL);
```

```
CREATE TABLE weapon
(wid INT PRIMARY KEY,
wname VARCHAR2(31) NOT NULL,
pdamage INT NOT NULL ,
mdamage INT NOT NULL ,
rarity varchar2(15) NOT NULL,
hpid INT NOT NULL,
bname VARCHAR2(31) NOT NULL,
qname VARCHAR2(31) NOT NULL);
```

```
CREATE TABLE area
(aname VARCHAR2(31) PRIMARY KEY,
asize NUMBER(10,2) NOT NULL,
restricted INT NOT NULL);
```

```
CREATE TABLE monster
(mname VARCHAR2(31) PRIMARY KEY,
mlevel INT NOT NULL,
aname VARCHAR2(31) NOT NULL);
```

```
CREATE TABLE MHP(
mlevel INT PRIMARY KEY,
mhp INT NOT NULL
```

);

```
CREATE TABLE WVALUE(  
rarity varchar2(31),  
damageM INT,  
damageP INT,  
VALUE INT NOT NULL,  
PRIMARY KEY(rarity,damageM,damageP)  
);
```

```
CREATE TABLE fights(  
hpid INT REFERENCES hero(pid),  
mname VARCHAR2(31) REFERENCES monster(mname),  
PRIMARY KEY(hpid,mname));
```

```
CREATE TABLE embarks_on(  
hpid INT REFERENCES hero(pid),  
qname VARCHAR2(31) REFERENCES quest(qname),  
PRIMARY KEY(hpid,qname));
```

```
CREATE TABLE takes_place_in(  
aname VARCHAR2(31) REFERENCES area(aname),  
qname VARCHAR2(31) REFERENCES quest(qname),  
PRIMARY KEY(aname, qname));
```

```
INSERT INTO hero VALUES (1, 'Barbarian', TO_DATE('01-12-2016','DD-MM-YYYY'), 0);  
INSERT INTO hero VALUES (2, 'DemonHunter', TO_DATE('02-11-2015','DD-MM-YYYY'), 0);  
INSERT INTO hero VALUES (3, 'Monk', TO_DATE('03-10-2014','DD-MM-YYYY'), 0);  
INSERT INTO hero VALUES (4, 'WitchDoctor', TO_DATE('04-09-2013','DD-MM-YYYY'), 0);
```

```
INSERT INTO backpack VALUES ('BarMoneyPack', 1, 20, 1000);  
INSERT INTO backpack VALUES ('DMWeaponPack', 2, 20, 2000);  
INSERT INTO backpack VALUES ('MonkTreasurePack', 3, 20, 3000);  
INSERT INTO backpack VALUES ('WDDDeadPack', 4, 20, 4000);  
INSERT INTO backpack VALUES ('BarPickPack', 1, 20, 1000);  
INSERT INTO backpack VALUES ('DMFoodPack', 2, 20, 2000);  
INSERT INTO backpack VALUES ('MonkPoisonPack', 3, 20, 3000);
```



INSERT INTO backpack VALUES ('WDHeadPack', 4, 20, 4000);

INSERT INTO quest VALUES ('Save the Princess', 1, 'Rescue', 'Medium');  
INSERT INTO quest VALUES ('Slay the Eternal Dragon', 0, 'Hunt', 'Insane');  
INSERT INTO quest VALUES ('Run for the Olympic', 1, 'Compete', 'Easy');  
INSERT INTO quest VALUES ('Create a country', 0, 'Train', 'Insane');  
INSERT INTO quest VALUES ('Recruit the Farmers', 1, 'Train', 'Medium');  
INSERT INTO quest VALUES ('The human should be killed', 0, 'Hunt', 'Easy');

INSERT INTO weapon VALUES (1, 'Buster Sword', 100, 5, 'Common', 1, 'BarMoneyPack',  
'Save the Princess');  
INSERT INTO weapon VALUES (2, 'Divine Chainsaw Blade', 50, 120, 'Magic', 2,  
'DMWeaponPack', 'Slay the Eternal Dragon');  
INSERT INTO weapon VALUES (3, 'Bow of Windforce', 200, 15, 'Rare', 3, 'MonkTreasurePack',  
'Run for the Olympic');  
INSERT INTO weapon VALUES (4, 'Holy Hammer', 10, 300, 'Unique', 4, 'WDDeadPack',  
'Create a country');  
INSERT INTO weapon VALUES (5, 'Andariels Visage', 200, 15, 'Rare', 1, 'BarPickPack', 'Recruit  
the Farmers');  
INSERT INTO weapon VALUES (6, 'Dead Mans Legacy', 10, 300, 'Unique', 2, 'DMFoodPack',  
'Create a country');  
INSERT INTO weapon VALUES (7, 'Chantodos Force', 200, 15, 'Rare', 3, 'MonkPoisonPack',  
'The human should be killed');  
INSERT INTO weapon VALUES (8, 'Weight of the Earth', 10, 300, 'Unique', 4, 'WDHeadPack',  
'Create a country');

INSERT INTO area VALUES ('Leviathan Peak', 1029.24, 0);  
INSERT INTO area VALUES ('Forbidden Forest', 465832.33, 1);  
INSERT INTO area VALUES ('Castle of Doom', 729.91, 0);  
INSERT INTO area VALUES ('Undead land', 203845.49, 1);  
INSERT INTO area VALUES ('Dragon Home', 534921.64, 0);  
INSERT INTO area VALUES ('Castle Vampire', 8901.64, 1);

INSERT INTO monster VALUES ('Dragonling', 20, 'Leviathan Peak');  
INSERT INTO monster VALUES ('Eternal Dragon', 31, 'Forbidden Forest');  
INSERT INTO monster VALUES ('Spooky Ghost', 42, 'Castle of Doom');  
INSERT INTO monster VALUES ('Golden Goblin', 53, 'Undead land');  
INSERT INTO monster VALUES ('Skeleton King', 64, 'Dragon Home');  
INSERT INTO monster VALUES ('Key warden', 75, 'Castle Vampire');  
INSERT INTO monster VALUES ('Bloody Thief', 21, 'Leviathan Peak');  
INSERT INTO monster VALUES ('Jumping Corpse', 32, 'Forbidden Forest');  
INSERT INTO monster VALUES ('Azure Drake', 43, 'Castle of Doom');  
INSERT INTO monster VALUES ('Berserker Slayer', 54, 'Undead land');

```
INSERT INTO monster VALUES ('Bonfire', 65, 'Dragon Home');
INSERT INTO monster VALUES ('Captain Stupid', 76, 'Castle Vampire');
```

```
INSERT INTO embarks_on VALUES (1, 'Save the Princess');
INSERT INTO embarks_on VALUES (2, 'Slay the Eternal Dragon');
INSERT INTO embarks_on VALUES (3, 'Run for the Olympic');
INSERT INTO embarks_on VALUES (4, 'Create a country');
INSERT INTO embarks_on VALUES (1, 'Run for the Olympic');
INSERT INTO embarks_on VALUES (2, 'Create a country');
INSERT INTO embarks_on VALUES (3, 'Recruit the Farmers');
INSERT INTO embarks_on VALUES (4, 'The human should be killed');
```

```
INSERT INTO takes_place_in VALUES ('Leviathan Peak', 'Save the Princess');
INSERT INTO takes_place_in VALUES ('Forbidden Forest', 'Slay the Eternal Dragon');
INSERT INTO takes_place_in VALUES ('Castle of Doom', 'Run for the Olympic');
INSERT INTO takes_place_in VALUES ('Undead land', 'Create a country');
INSERT INTO takes_place_in VALUES ('Dragon Home', 'Recruit the Farmers');
INSERT INTO takes_place_in VALUES ('Castle Vampire', 'The human should be killed');
INSERT INTO takes_place_in VALUES ('Castle Vampire', 'Create a country');
```

```
CREATE TABLE consumable
(cname VARCHAR2(31),
hpid INT REFERENCES hero(pid),
bname VARCHAR2(31),
PRIMARY KEY(hpid,bname,cname));
```

```
INSERT INTO consumable VALUES ('Magical Soup', 1, 'BarMoneyPack');
INSERT INTO consumable VALUES ('Poison Apple', 2, 'DMWeaponPack');
INSERT INTO consumable VALUES ('Blessing of prince', 3, 'MonkTreasurePack');
INSERT INTO consumable VALUES ('Gods mercy', 4, 'WDDeadPack');
INSERT INTO consumable VALUES ('Devil Tail', 1, 'BarMoneyPack');
INSERT INTO consumable VALUES ('Rabbit meat', 3, 'MonkPoisonPack');
INSERT INTO consumable VALUES ('Piggy Bottom', 1, 'BarPickPack');
INSERT INTO consumable VALUES ('Angry Duck', 2, 'DMFoodPack');
```

```
commit;
```

## Part 5: Queries

**(1). Retrieve the name of quests whose qname contains key word “Dragon”**

```
SELECT qname
FROM quest
WHERE qname like '%Dragon%';
```

**(2). List the name of the hero whose birthday is 04-09-2013**

```
SELECT hname
FROM hero
WHERE bdate = TO_DATE('04-09-13','DD-MM-YY');
```

**(3). Classify the quests by difficulty, Then list difficulty and the number of quests only when there are more than one quests in the group.**

```
SELECT difficulty,COUNT(qname)
FROM quest
GROUP BY difficulty Having COUNT(qname) > 1;
```

**(4). List the area whose size is larger than their average.**

```
SELECT DISTINCT *
FROM area outer
WHERE outer.asize >
      (SELECT AVG(inner.asize)
       FROM area inner);
```

**(5). List all the player that has at least one backpack.**

```
SELECT DISTINCT pid,hname
FROM hero e
WHERE EXISTS
      (SELECT *
       FROM backpack b
       WHERE b.hpid=e.pid);
```

**(6). List the number of heros in the database.**

```
SELECT COUNT(pid)
FROM hero;
```

**7). Delete all the consumable items that belong to player id 1.**

```
delete from consumable where hpid=1;
```

**(8). List the hero, the quests that they've embarked upon, and the weapons they've been rewarded with.**

```
SELECT hero.hname,embarks_on.qname,weapon.wname  
FROM weapon,hero,embarks_on  
WHERE hero.pid=embarks_on.hpid and weapon.qname=embarks_on.qname;
```

## **Part 6: Teamwork Summary**

Overall, the work was evenly spread out. Each task was more or less shared among the group members but everyone had a major role to play in a certain area.

Thomas worked on the normalisation process.

Jialin looked over and proofread all queries, processes and normalisation.

Robert worked on turning the tables from BCNF form to SQL.

Gao worked on inserting fake data into the database.

Sonny worked on queries for the database.

Amy worked on the conversion from ERD to relational schema and documented the process.