Databases Programming Project: **Final** **Report**

이름 (Name): 정진택

학과 (Department): 정보컴퓨터공학부

학번 (Student ID): 202155655

1. **프로젝트 개요 (Project topic)**

|  |
| --- |
| ※ 프로젝트 주제, 목표와 본인 프로젝트의 차별성, 장점 등에 대해 상세히 언급해주세요.  사용한 언어 및 라이브러리: python, pycharm, psycopg2  프로젝트 주제: 인간 키우기 – 전투 RPG  목표:   1. **캐릭터 관리 시스템:** 사용자가 게임 내에서 생성한 가상의 인간 캐릭터를 데이터베이스에 등록하고, 캐릭터의 레벨, 경험치, 보유 장비 등의 정보를 효율적으로 관리합니다. 2. **아이템 및 장비 관리 시스템:** 게임 내의 다양한 아이템과 장비를 데이터베이스에서 관리하여, 사용자가 획득하거나 장착한 아이템에 대한 정보를 정확하게 추적합니다. 3. **전투 및 몬스터 관리 시스템:** 사용자가 몬스터와의 전투를 진행할 때, 인간 정보와 몬스터 정보를 데이터베이스에 기록하고, 몬스터의 생성, 삭제, 전투 결과 등을 체계적으로 관리합니다.   장점 및 차별성:   1. **실제 게임 데이터 모델링:** 마인크래프트 게임 내 캐릭터, 아이템, 몬스터 등의 데이터를 모델링하고 이를 효율적으로 저장, 관리합니다. 이를 통해 실제 게임에서 사용되는 일부 데이터베이스와 유사한 구조를 구현합니다. 2. **데이터베이스의 CRUD 연산 활용:** 사용자가 게임 내에서 이루어지는 다양한 활동에 대한 데이터베이스의 생성(Create), 조회(Read), 갱신(Update), 삭제(Delete) 연산을 적절히 활용하여 데이터 일관성과 효율성을 유지합니다. 예를 들어 몬스터와 인간 간의 전투에서 몬스터 처치 시 인간의 경험치 및 돈 증가, 무기 내구도 감소 등의 상황에서도 데이터베이스는 일관된 상태를 유지하며 오류 없는 정보 갱신이 이루어집니다. 3. **개발자 모드**: 사용자가 직접 몬스터를 생성하거나 다양한 게임 내 기능을 조작할 수 있는 개발자 모드를 도입하였습니다. 이로써 게임 내에서 다양한 몬스터와 무기 생성이 가능해집니다. 4. **동적인 뷰**: View를 활용하여 복잡한 쿼리가 간소화되고 사용자에게 가독성 높은 정보를 제공합니다. 사용자는 여러 테이블 간의 복잡한 조인을 몰라도 뷰를 통해 쉽게 필요한 정보에 접근할 수 있습니다. 특히, monster\_attack\_info 뷰를 통해 사용자의 몬스터 공격 정보를 한눈에 확인할 수 있습니다. 이를 통해 사용자는 게임 진행 상황을 실시간으로 파악하고 조작할 수 있습니다. 5. **권한:** 사용자에게 테이블에 대해 authorization을 인가하여 각 권한이 어떤 행동에 사용되는지 투명하게 관리할 수 있습니다. 6. **아이템 및 몬스터 중복 생성 방지**: 아이템과 몬스터를 생성할 때 중복을 방지하기 위해 데이터베이스 내에서 유니크한 조건을 설정합니다. 또한 함수 내에서 각 아이디를 생성할 때 id값이 중복되지 않도록 로직을 구성하였습니다. 이를 통해 사용자는 중복된 아이템이나 몬스터로 인한 불필요한 상황을 경험하지 않습니다. 7. **입력 오류 처리 및 안내**: 사용자가 부적절한 입력을 할 경우에 대한 오류 처리 로직을 구성하였습니다. 사용자가 예상치 못한 입력을 했을 때, 명확한 안내 메시지와 함께 예외 처리를 통해 시스템을 안정적으로 유지합니다. 8. **동시성 관리와 트랜잭션 활용:** 몬스터 생성, 아이템 생성, 인간 티어 승급, 인벤토리 조작, 인간 정보 갱신 등의 기능에서 트랜잭션을 사용하여 여러 동작이 동시에 수행되더라도 일관성을 유지하고 데이터의 무결성을 보장하였습니다. 9. **티어:** 티어를 도입하여 사용자가 캐릭터를 성장시키는데 재미를 붙일수 있도록 하였습니다. 10. **회원가입:** 회원가입과 로그인 기능이 있어, 사용자가 다양한 계정을 생성하고 게임을 즐길 수 있습니다. |

1. **사용자 (역할) (Users / Roles)**

|  |
| --- |
| 인간 키우기 게임에는 인간(human), 몬스터(monster), 무기(weapon), 물약(potion), 상점(item\_shop), 창고(inventory), 그리고 개발자(developer) 역할이 존재합니다.   * **인간 (Human):**   + 행동: 몬스터를 공격하여 돈과 경험치를 얻음, 돈을 쓰고 상점에서 무기와 물약을 구매함. 경험치를 얻으면 자동 승급함.   + 특징: 몬스터에게 공격당하며, 몬스터를 처치하면 돈과 경험치 획득 * **몬스터 (Monster):**   + 행동: 인간에게 공격을 가하고 동시에 공격을 받음   + 결과: 몬스터 처치 시, 인간에게 돈과 경험치 제공 * **무기 (Weapon):**   + 활용: 돈을 모아 무기를 구매하여 인간의 공격력 향상   + 특징: 몬스터를 공격할 때마다 내구도가 감소 * **물약 (Potion):**   + 활용: 돈을 사용해 물약을 구매하여 인간의 체력 회복   + 섭취: 물약을 인간이 섭취하여 체력 회복 * **상점 (Item Shop):**   + 제공: 무기와 물약을 판매   + 저장: 구매한 아이템은 창고에 자동 저장 * **창고 (Inventory):**   + 보관: 구매한 무기와 물약이 자동으로 창고에 저장   + 기능: 무기는 장착 및 해제가 가능하며, 물약은 섭취 가능 * **개발자 (Developer):**   + 기능: 몬스터 생성, 아이템 생성 등의 게임 기능을 개발자가 수행   이 게임에서는 계정당 하나의 인간 캐릭터가 생성되며, 플레이어는 무기와 물약을 구매하여 캐릭터를 강화하고 몬스터를 처치해 돈과 경험치를 얻습니다. 개발자는 게임의 기능을 개선하거나 새로운 콘텐츠를 추가하는 역할을 수행합니다. |

1. **기능 (Functions)**

|  |
| --- |
| ※ 각 기능이 어떤 사용자를 위한 것이며, 어떤 SQL feature를 사용하는지 명시해주세요.  인간 (Human)   * 회원가입: 계정을 생성함. * 사용자에게 id와 password를 입력받음. * id 중복 여부를 확인하고, 중복된 값이 아니라면 계정을 생성함. * Game\_account, human 테이블에 정보를 추가함. * 로그인: 생성한 계정으로 게임에 접속함. * 사용자에게 id와 password를 입력받음. * Id와 password가 일치하면 로그인하고, 일치하지 않으면 다시 로그인하라는 메시지 출력과 동시에 로그인 실패. * 계정 상태 보기: 생성되어있는 계정들의 상태를 확인함. * Game\_account table에서 id, name을 조회하여 출력함. * 생성된 몬스터 상태 보기: 생성된 몬스터의 상태정보를 보여줌. * Created\_monster에서 monster\_id, monster\_name, attack, hp, reward\_exp, reward\_money를 조회하여 출력한다. * 공격할 몬스터 지정: 공격할 몬스터를 지정함. * 사용자가 입력한 monster\_id가 created\_monster 중 하나의 monster\_id와 일치한지 확인한다. * 일치하면 Monster\_attack에서 human\_id와 monster\_id에 등록한다. * 일치하지 않으면 등록실패. * 다른 사용자가 공격중인 몬스터일 경우에도 등록실패 * 이미 한 몬스터를 공격하고 있더라도 다른 몬스터를 다시 지정할 수 있다. 이때 원래 공격받던 몬스터는 삭제된다. * 몬스터 공격: 몬스터를 공격하고 동시에 공격받음. * Monster\_attack\_info라는 view를 활용하여 monster\_attack에서 human이 공격중인 몬스터를 찾고 몬스터와 human 정보를 가져온다. * 인간의 attack만큼 몬스터의 hp가 감소하고, 몬스터의 attack만큼 인간의 hp가 감소한다. * 몬스터의 hp가 0이 되면 몬스터가 죽으면서 인간에게 경험치와 돈을 준다. * 인간의 경험치를 보고 rank테이블에서 비교하여 자동승급 시켜준다. * 인간의 hp가 0이 되면 죽고, 자동으로 로그아웃되면서 human과 game\_account에서 해당 계정에 대한 정보가 삭제된다. 위 과정에서 dml이 적용된다. * 캐릭터 상태 보기: 인간의 상태정보를 확인함. * 인간의 human\_id, human\_name, attack, hp, exp, money,   equipped\_weapon, rank 를 조회하여 출력한다.  창고 (Inventory)   * 아이템 정보 보기: 현재 inventory에 있는 item들을 확인함. * Inventory에 있는 item\_name, item\_id, item\_type 를 조회하여 출력한다. * 무기 정보 보기: 인간이 가진 무기 정보 확인. * Inventory\_weapons\_view라는 view를 활용하여 Human\_weapon에서 item\_id가 일치하는 컬럼의 weapon\_id, weapon\_name, attack, duration을 조회하여 출력함. * 물약 정보 보기: 인간이 가진 물약 정보 확인. * Inventory\_potions\_view라는 view를 활용하여 Human\_potion에서 item\_id가 일치하는 컬럼의 potion\_id, potion\_name, count, heal을 조회하여 출력함. * 무기 장착: inventory에서 무기를 꺼내 human이 장착함. * 사용자에게 장착할 무기 id를 입력받는다. * 해당 무기가 인벤토리에 있다면 무기를 장착한다. * 무기 attack만큼 인간이 attack을 갖는다. * 이후 인간이 무기를 사용하여 공격하면 무기의 내구도는 감소한다. * 무기 해제: human이 장착한 무기를 장착 해제함. * 현재 인간이 장착한 무기가 있다면 해제한다. * 인간의 attack과 equipped\_weapon\_id를 다시 기본값으로 돌려놓는다. * 물약 섭취: human이 물약을 먹음. * 섭취할 물약의 id와 개수를 입력받는다. * 인벤토리에 해당 물약이 입력받은 개수 이상으로 존재한다면 물약을 섭취한다. * 개수가 부족하거나 창고에 없는 물약이라면 물약섭취에 실패한다. * 물약 섭취후에는 인간의 hp가 회복되고, inventory와 human\_potion에서 물약이 해당개수만큼 제거된다.   상점(Item\_shop)   * 상점 보기: 사용자에게 상점에 있는 item들을 보여줌. * Item\_shop에서 item\_name과 item\_type을 조회하여 출력한다. * 아이템 정보 확인: 사용자가 고른 item의 정보를 찾아줌. * 사용자가 고른 item\_name을 item\_type에 따라 Weapon과 Potion에서 일치하는 값을 찾음. * 일치하는 컬럼의 정보들을 출력함. * Item 판매: 사용자가 상점에서 item을 구매할 수 있음. * 구매할 아이템 이름을 입력받는다. * 사용자가 고른 아이템의 가격을 조회한다. * 인간의 현재 money를 조회한다. * Money 가 price보다 작으면 아이템 판매에 실패한다. * 인간의 money가 가격만큼 감소하고, 무기를 구입할경우, inventory와 human\_weapon에 정보가 추가되고, 물약을 구입할경우, inventory와 human\_potion에 정보가 추가된다. * Human\_potion에 이미 있는 name일 경우, 해당 potion의 count만 증가시킨다.   개발자(developer)   * 개발자 생성: 개발자가 없는 게임 초기에 개발자 id를 자동생성함. * Developer 테이블에 데이터가 없을 경우 id가 0인 developer를 생성함. * 이후로 다른 id는 생성하지 않음. * Game\_account에도 추가함. * 몬스터 생성: 몬스터를 생성함. * 생성할 몬스터의 번호와 개수를 입력받는다. * 각기 다른 monster\_id를 가진 몬스터들이 created\_monster에 추가된다. * 몬스터 정보 보기: 생성가능한 몬스터들의 정보를 보여줌. * Monster 테이블에서 각 몬스터의 name, attack, hp, reward\_exp와 reward\_money를 조회하여 출력한다. * Item 생성: item들을 생성하고 상점에 item들을 등록함. * 생성할 아이템의 name과 type을 입력받는다. * Type이 weapon이라면 weapon의 attack, price, durability를 입력받는다. * Type이 potion이라면 potion의 price와 회복력(heal)을 입력받는다. * item\_shop에 데이터를 추가하고, weapon이라면 weapond에 potion이라면 potion에 데이터를 추가한다. * Item 제거: 생성한 item들을 제거하고 상점에 등록된 item들도 제거함. * Item\_shop에서 item\_name과 item\_type들을 조회하고 출력함. * 사용자가 고른 item\_name에 맞는 아이템을 item\_shop에서 제거함. * Item\_type을 item\_shop에서 조회하여 weapon일 경우 weapon에서 해당 item을 제거함. * Item\_type이 potion일 경우 potion에서 해당 item을 제거함. * 몬스터 제거: 생성한 몬스터를 제거함. * Created\_monster에서 개발자가 입력한 monster\_id를 찾고 해당 컬럼을 삭제함. * 찾지못하면 제거 실패.   - rank 추가: rank테이블에 정보 추가함.   * 생성할 rank와 필요한 exp를 입력받음. * rank테이블에 데이터를 삽입함.   - rank 제거: rank테이블에 정보 제거함.   * Rank테이블에서 rank를 조회하여 출력함. * 사용자가 고른 rank를 rank테이블에서 삭제함. |

1. **데이터베이스 스키마 및 다이어그램 (Database schema / Schema diagram)**

|  |
| --- |
| Human(human\_id, human\_name, attack, hp, exp, money, equipped\_weapon\_id, rank, )   * Foreign key: human\_id -> Game\_account(id), human\_name ->Game\_account(name), rank -> rank(rank) * Human\_id -> Monster\_attack(human\_id) * id: 인간의 id를 var\_char형으로 나타낸다. * attack: 인간의 공격력 정보를 int형으로 나타낸다. 무기가 없을 때 기본공격력인 2 를 갖는다. * hp: 인간의 체력 정보를 int형으로 나타낸다. 최대 체력은 100이다. 기본값은 100이다. * exp: 인간의 경험치 정보를 int형으로 나타낸다. 0 이상의 값을 갖는다. 기본값은 0이다. * money: 인간이 현재 가지고 있는 금액을 int형으로 나타낸다. 0 이상의 값을 갖는다. 기본값은 0이다. * Equipped\_weapon: 인간이 지니고 있는 무기를 var\_char형으로 나타낸다. 무기에 따라 attack이 변한다. 무기는 inventory에서 꺼내서 장착/해제가 가능며, 하나씩만 장착가능하다. 처음엔 null 값을 갖는다. * rank: 인간의 현재 계급을 var\_char형으로 나타낸다. 기본값은 null이다.   Developer(id, login\_state)   * id: 개발자 모드 id를 var\_char형으로 나타낸다. 이 값은 하나뿐이다. * Login\_state: 개발자 모드 접속상태를 int형으로 나타낸다.   Monster\_attack(human\_id, monster\_id)   * Foreign Key: monster\_id -> created\_monster(monster\_id) * human\_id: 공격 중인 human의 id를 var\_char형으로 나타낸다. * Monster\_id: 공격 받고 있는 monster의 id를 var\_char형으로 나타낸다.   Game\_account(id, name, password)   * id: 계정의 id를 var\_char형으로 나타낸다. 이 id는 사용자가 회원가입시에 입력하게되는데, 이 값은 생성되는 캐릭터인 human의 id와 같다. 이 값은 고유하다. 개발자 id는 0으로 저장된다. * Name: 계정의 name을 var\_char형으로 나타낸다. 이 값은 사용자가 회원가입시에 입력하게되는데, 이 값은 생성되는 캐릭터인 human의 name과 같다. 이 값은 고유하다. 개발자 name은 developer로 저장된다. * Password: 계정의 비밀번호를 var\_char형으로 나타낸다. 개발자 password는 0으로 저장된다.   Human\_weapon(weapon\_id, weapon\_name, attack, durability)   * Human이 소유하는 무기의 정보를 나타낸다. * Weapon\_id: 무기의 id를 var\_char형으로 나타낸다. 이 값은 중복되지않는다. * weapon\_name: 무기의 이름을 var\_char형으로 나타낸다. * attack: 무기의 공격력을 int형으로 나타낸다. 0 이상의 값을 갖는다. * durability: 무기의 내구도를 int형으로 나타낸다. 0 이상의 값을 갖는다.   Human\_potion(Potion\_id, Potion\_name, count, heal)  Human이 소유하는 포션의 정보를 나타낸다.   * potion\_id: 물약의 id를 var\_char형으로 나타낸다. 이 값은 중복되지않는다. * Potion\_name: 물약의 이름을 var\_char형으로 나타낸다. * Count: 물약의 개수를 int형으로 나타낸다. 0 이상의 값을 갖는다. * heal: 물약 1개당 hp회복능력을 int형으로 나타낸다.   Weapon(weapon\_name, attack, price, durability)  상점에 있는 무기의 정보를 나타낸다.   * ForeignKey: weapon\_name->item\_shop(item\_name) * weapon\_name: 무기의 이름을 var\_char형으로 나타낸다. * attack: 무기의 공격력을 int형으로 나타낸다. 0 이상의 값을 갖는다. * Price: 무기를 구입하는데 드는 money를 int형으로 나타낸다. * durability: 무기의 내구도를 int형으로 나타낸다. 0 이상의 값을 갖는다.   Potion(Potion\_name, price, heal)  상점에 있는 포션의 정보를 나타낸다.   * ForeignKey: potion\_name->item\_shop(item\_name) * Potion\_name: 물약의 이름을 var\_char형으로 나타낸다. * price: 물약의 가격을 int형으로 나타낸다. * Heal: 물약의 회복능력을 int형으로 나타낸다.   Inventory(human\_id, item\_id, item\_name, item\_type)   * ForeignKey: human\_id->Human(human\_id),   item\_name->Item\_shop(item\_name)   * Human\_id: 아이템 소유자의 id를 var\_char형으로 나타낸다. * Item\_id: 아이템의 id를 var\_char형으로 나타낸다. * Item\_name: 아이템의 name을 var\_char형으로 나타낸다. * Item\_type: 아이템의 종류를 var\_char형으로 나타낸다.   Item\_shop(item\_name, item\_type)   * Item\_name: 상점에서 파는 item들의 이름을 var\_char형으로 나타낸다. * Item\_type: 상점에서 파는 item들의 type을 var\_char형으로 나타낸다.   Monster(monster\_name, attack, hp, reward\_exp, reward\_money)   * monster\_name: 몬스터의 이름을 var\_char형으로 나타낸다. * attack: 몬스터의 공격력을 int형으로 나타낸다. Attack은 0이상의 값을 갖는다. * hp: 몬스터의 체력을 int형으로 나타낸다. hp는 0이상의 값을 갖는다. * reward\_exp: 인간이 몬스터를 잡을 때 얻을 수 있는 경험치를 int형으로 나타낸다. 0 이상의 값을 갖는다. * reward\_money: 인간이 몬스터를 잡을 때 얻을 수 있는 금액을 int형으로 나타낸다. 0 이상의 값을 갖는다.   Created\_monster(monster\_id, monster\_name, attack, hp, reward\_exp, reward\_money)   * ForeignKey: monster\_name->Monster(monster\_name) * Monster\_id: 생성된 몬스터의 id값을 var\_char형으로 나타낸다. * Monster\_name: 생성된 몬스터의 이름을 var\_char형으로 나타낸다. * attack: 생성된 몬스터의 공격력을 int형으로 나타낸다. Attack은 0이상의 값을 갖는다. * hp: 생성된 몬스터의 체력을 int형으로 나타낸다. * reward\_exp: 인간이 몬스터를 잡을 때 얻을 수 있는 경험치를 int형으로 나타낸다. 0 이상의 값을 갖는다. * reward\_money: 인간이 몬스터를 잡을 때 얻을 수 있는 금액을 int형으로 나타낸다. 0 이상의 값을 갖는다.   Rank(rank, exp)   * Rank: 티어를 var\_char형으로 나타낸다. * Exp: 티어별 필요한 경험치를 int형으로 나타낸다. |

1. **(팀 프로젝트인 경우만 해당) 팀원의 역할 배분**

|  |
| --- |
| ※ 각 팀원이 수행한 업무를 기재해주세요. |