University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 02/27/2024

Group Number: 90

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Desheng Lin	29605730	z3m1r	deshenglin127@gmail.com
James Wang	62824172	e0v7t	james.wang.0303@gmail.com
Xiran Wei	27327246	u5v5o	ommjfk@gmail.com

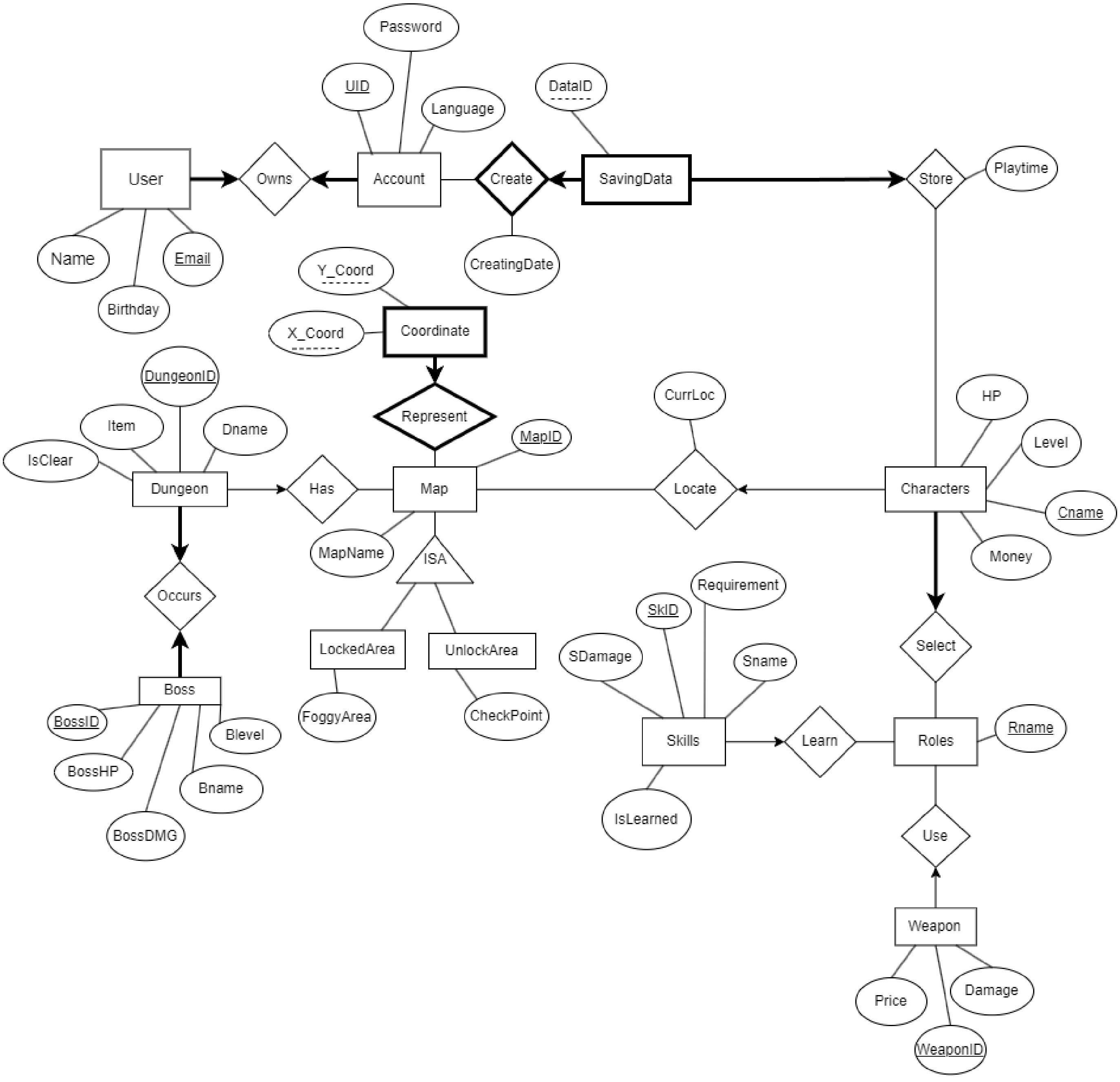
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

2. Brief Summary

Our project is focused on building a game database system where users can create accounts, select characters with specific roles and skills, and engage in boss fights within a map, potentially including dungeons. The database, implemented using the Oracle database system, models entities such as user information, accounts, characters, and map details, along with their relationships. The system allows users to store and modify in-game data, while game developers can make updates to introduce new features, roles, and bosses.

3. ER Diagram



Changes to ERD:

- UserInfo is no longer a Weak Entity of Account.
- Password is no longer a Primary Key of Account.
- Roles is no longer a Weak Entity of Characters.
- Learn is now a Relationship between Roles and Skills.
- ISA Relationship has been deleted from Skills.
    - IsLearn, a new attribute, has been added to Skills.

4. Schema

For the following schemas, underlined attributes denote primary key attributes, and **bold** attributes denote foreign key attributes.

- UserOwnsAccount(Email: VARCHAR, Name: VARCHAR, Birthday: DATE, UID: int(10), Password: VARCHAR, Language VARCHAR)
  Language: NOT NULL
  Password: NOT NULL
  Name: NOT NULL
  UID: Candidate Key
- SaveCreateStoreData(DataID: CHAR(10), CreatingDate: DATE, **UID**: int(10), playTime: VARCHAR, **Cname**: VARCHAR)
  CreatingDate: NOT NULL
- Characters(Cname: VARCHAR, Level: INT(2), HP: INT(10), Money: INT(10), **Rname**: VARCHAR, **MapID**: CHAR(10), currLoc: CHAR(20))
  Level: NOT NULL
  HP: NOT NULL

Money: NOT NULL
currLoc: NOT NULL

- Roles(Rname: VARCHAR)

- Weapon(Wid: char(10), Damage: int(10), Price: int(10), **Rname**: char(20))
  Damage: NOT NULL

- learnSkills(SDamage: int(10), SkID: char(10), Sname: VARCHAR, Requirement: VARCHAR, Islearned: boolean, **Rname**: VARCHAR)
  Sname: UNIQUE
  Islearned: NOT NULL

- LockedArea(MapID: char(10), Mname: VARCHAR, FoggyArea: char(10))
  Mname: NOT NULL & UNIQUE

- UnlockArea(CheckPoint: char(10), MapID: char(10), Mname: VARCHAR)
  Mname: NOT NULL & UNIQUE

- Coordinate(Y_Coord: char(10), Y_Coord: char(10), **MapID**: char(10))

- DungeonOccurBoss (DungeonID: char(10), Dname: VARCHAR, Item: VARCHAR, IsClear: boolean, BLevel: int(2), Bname:

VARCHAR, BossDMG: int(10), BossHP: int(10), BossID: char(10), **MapID**: char(10))
Dname: UNIQUE & NOT NULL
IsClear: NOT NULL
BLevel: NOT NULL
Bname: UNIQUE & NOT NULL
BossID: Candidate Key

5. Functional Dependencies

- UserOwnsAccount(Email: VARCHAR, Name: VARCHAR, Birthday: date, UID: int(10), Password: VARCHAR, Language VARCHAR)
  - Email→ Name, Birthday, UID, Password, Language
  - UID→ Name, Birthday, Password, Language, Email
- SaveCreateStoreData(DataID: char(10), CreatingDate: date, UID: int(10), Playtime: VARCHAR, **Cname**: VARCHAR)

  DataID, UID→ CreatingDate, Playtime, Cname
- Characters(Cname: VARCHAR, Level: int(2), HP: int(10), Money: int(10), **Rname**: VARCHAR, **MapID**: char(10), currLoc: char(20))
  - Cname→HP, **MapID**, Rname, currLoc, Money
  - Level→HP
- Weapon(Wid: char(10), Damage: int(10), Price: int(10), **Rname**: char(20))

  WeaponID → Damage, Price, Rname
- LearnSkills(SDamage: int(10), SkID: char(10), Sname: VARCHAR, Requirement: VARCHAR, Islearned: boolean, **Rname**: VARCHAR)
  - SkillID→SDamage, Sname, Requirement, IsLearned, **Rname**
  - Sname→SDamage, Requirement

- LockedArea(MapID: char(10), Mname: VARCHAR, FoggyArea: char(10))
    - MapID → MapName, FoggyArea
- UnlockArea(CheckPoint: char(10), MapID: char(10), Mname: VARCHAR)
    - MapID → MapName, CheckPoint
- DungeonOccurBoss (DungeonID: char(10), Dname: VARCHAR, Item: VARCHAR, IsClear: boolean, BossLevel: int(2), Bname: VARCHAR, BossDMG: int(10), BossHP: int(10), BossID: char(10), **MapID**: char(10))
    - DungeonID → DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, BossID, **MapID**
    - BossID→ DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, DungeonID, MapID
    - BossName→ BossLevel, BossDMG, BossHP
    - DungeonName→ ClearStatus

6. Normalization

We will use the lossless-join BCNF decomposition to ensure the relations follow the BCNF specification.

The tables that violate BCNF are:

- Characters(Cname: VARCHAR, Level: int(2), HP: int(10), Money: int(10), **Rname**: VARCHAR, **MapID**: char(10), currLoc: char(20))
    - Cname→HP, **MapID**, **Rname**, currLoc, Money
    - Level→HP
- LearnSkills(SDamage: int(10), SkillID: char(10), Sname: VARCHAR, Requirement: VARCHAR, Islearned: boolean, **Rname**: VARCHAR)

    - SkillID→SDamage, Sname, Requirement, IsLearned, **Rname**
    - Sname→SDamage, Requirement
- DungeonOccurBoss (DungeonID: char(10), Dname: VARCHAR, Item: VARCHAR, IsClear: boolean, BossLevel: int(2), Bname: VARCHAR, BossDMG: int(10), BossHP: int(10), BossID: char(10), **MapID**: char(10))
    - DungeonID → DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, BossID, **MapID**
    - BossID→ DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, DungeonID, **MapID**
    - BossName→ BossLevel, BossDMG, BossHP
    - DungeonName→ ClearStatus

Decomposition:

- Characters(Cname: VARCHAR, Level: int(2), HP: int(10), Money: int(10), **Rname**: VARCHAR, **MapID**: char(10), currLoc: char(20))
    - Cname→HP, **MapID**, **Rname**, currLoc, Money
    - Level→HP

Cname is the PK

The closure for Level+ = {Level, HP}

Decompose Level→HP

The new tables are:

- Characters_Stats(Level: int(2), HP: int(10))
- Characters_Info(Cname: VARCHAR, Level: int(2), Money: int(10), **Rname**: VARCHAR, **MapID**: char(10), currLoc: char(20))

- LearnSkills(SDamage: int(10), SkillID: char(10), Sname: VARCHAR, Requirement: VARCHAR, Islearned: boolean, **Rname**: VARCHAR)

    - SkillID→SDamage, Sname, Requirement, IsLearned, **Rname**
    - Sname→SDamage, Requirement

SkillID is the PK
The closure for Sname+ = {Sname, SDamage, Requirement}
Decompose Sname→SDamage, Requirement
The new tables are:

- LearnSkills_Stats(SDamage: int(10), Sname: VARCHAR, Requirement: VARCHAR)
- LearnSkills_Info(SkillID: char(10), Sname: VARCHAR, IsLearned: boolean, **Rname**: VARCHAR)

- DungeonOccurBoss (DungeonID: char(10), Dname: VARCHAR, Item: VARCHAR, IsClear: boolean, BossLevel: int(2), Bname: VARCHAR, BossDMG: int(10), BossHP: int(10), BossID: char(10), **MapID**: char(10))
    - DungeonID → DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, BossID, MapID
    - BossID→ DungeonName, Item, ClearStatus, BossLevel, BossName, BossDMG, BossHP, DungeonID, MapID
    - BossName→ BossLevel, BossDMG, BossHP
    - DungeonName→ ClearStatus

DungeonID is the PK
BossID is a Candidate key
The closure for BossName+ = {BossName, BossLevel, BossDMG, BossHP}
The closure for DungeonName+ = {DungeonName, ClearStatus}
Decompose BossName→ BossLevel, BossDMG, BossHP

- BossInfo (BossLevel: int(2), BossName: VARCHAR, BossDMG: int(10), BossHP: int(10))

- DungeonOccurBoss_2 (DungeonID: char(10), DungeonName: VARCHAR, Item: VARCHAR, ClearStatus: boolean, BossName: VARCHAR), BossID: char(10), **MapID**: char(10))

Decompose DungeonName→ ClearStatus in DungeonOccurBoss_2

- DungeonStats (DungeonName: VARCHAR, ClearStatus: boolean)
- DungeonInfo (DungeonID: char(10), DungeonName: VARCHAR, Item: VARCHAR, BossName: VARCHAR, BossID: char(10), **MapID**: char(10))

Therefore, the resulting tables are:

- BossInfo (BossLevel: int(2), BossName: VARCHAR, BossDMG: int(10), BossHP: int(10))
- DungeonStats (DungeonName: VARCHAR, ClearStatus: boolean)
- DungeonInfo (DungeonID: char(10), DungeonName: VARCHAR, Item: VARCHAR, BossName: VARCHAR, BossID: char(10), **MapID**: char(10))

7. SQL DDL

```
CREATE TABLE UserOwnsAccount(
Name VARCHAR,
Birthday DATE,
Email VARCHAR,
Password INTEGER,
Uid INTEGER,
UNIQUE(UID)
PRIMARY KEY(Email, Password, UID),
FOREIGN KEY(UID, Password))
```

```sql
CREATE TABLE SaveCreateStoreData(
DataID INTEGER,
UID INTEGER,
playTime DATE
creatingDate DATE,
Cname VARCHAR,
PRIMARY KEY(DataID, UID),
FOREIGN KEY(UID, Password),
FOREIGN KEY(Cname) REFERENCES Character(Cname) on DELETE CASCADE)

CREATE TABLE Characters_Stats(
HP INTEGER,
Level INTEGER PRIMARY KEY)

CREATE TABLE Characters_Info(
Level INTEGER,
Money INTEGER,
Cname VARCHAR PRIMARY KEY,
Rname VARCHAR,
MapID CHAR(10),
currLoc CHAR(20),
FOREIGN KEY(MapID) REFERENCES LockedArea(MapID),
UnlockArea(MapID)
FOREIGN KEY(Rname) REFERENCES Roles(Rname))

CREATE TABLE Weapons(
WeaponID CHAR(10),
wpDamage INTEGER,
Price INTEGER,
Rname CHAR(20),
FOREIGN KEY(Rname) REFERENCES Roles(Rname) on UPDATE
CASCADE))

CREATE TABLE Roles(
Rname CHAR(25),
Cname CHAR(25),
UNIQUE Rname
FOREIGN KEY(Cname) REFERENCES Character(Cname) on
UPDATE CASCADE))

CREATE TABLE LockedArea(
MapID CHAR(10),
MapName VARCHAR,
FoggyArea CHAR(10),
PRIMARY KEY(MapID))
CREATE TABLE UnlockArea(
MapID CHAR(10),
MapName VARCHAR,
CheckPoint CHAR(10),
PRIMARY KEY(MapID))

CREATE TABLE UnlockArea(
MapID CHAR(10),
MapName VARCHAR,
CheckPoint CHAR(10),
PRIMARY KEY(MapID))

CREATE TABLE Coordinate(
X_Coord CHAR(10),
MapID CHAR(10),
Y_Coord CHAR(10),
PRIMARY KEY(X_Coord,Y_Coord, MapID)
FOREIGN KEY(MapID) REFERENCES LockedArea(MapID),
UnlockArea(MapID))

CREATE TABLE LearnSkills_Stats(
Sname VARCHAR,
SDamage INTEGER,
Requirement VARCHAR,
PRIMARY KEY(Sname))

CREATE TABLE LearnSkills_Info(
skillID CHAR(10) PRIMARY KEY,
Sname VARCHAR,
IsLearned BOOLEAN,
Rname VARCHAR
FOREIGN KEY(Rname) REFERENCES Roles(Rname) on UPDATE
CASCADE))

CREATE TABLE BossInfo(
Bname VARCHAR,
Blevel INTEGER,
BossDMG INTEGER,
BossHP INTEGER,
PRIMARY KEY(Bname))
CREATE TABLE DungeonStats(
dungeonName VARCHAR PRIMARY KEY,
clearStatus BOOLEAN)
```

```
CREATE TABLE DungeonInfo(
dungeonID CHAR(10)  PRIMARY KEY,
dungeonName VARCHAR,
Item VARCHAR,
Bname VARCHAR,
BossID VARCHAR,
MapID CHAR(10),
FOREIGN KEY(MapID) REFERENCES LockedArea(MapID),
UnlockArea(MapID))
```

8. Populate tables

```
INSERT
INTO UserOwnsAccount (name, Birthday, Email, Password, Uid)
VALUES
        ('Bob', '2024-02-01', 'bob@student.ubc.ca', 'Bob', U000000304),
         ('Alice', '2024-02-01', 'alice@student.ubc.ca', 'Alice',
        U000000305),
        ('Andrew', '2024-02-01', 'andrew@student.ubc.ca', 'Andrew',
        U000000315),
         ('Billy', '2024-02-01', 'billy@student.ubc.ca', 'Billy', U000000306),
         ('Carol', '2024-02-01', 'carol@student.ubc.ca', 'Carol',
         U000000307),
         ('Lucy', '2024-02-01', 'lucy@student.ubc.ca', 'Lucy', U000000308);
```

```
INSERT
INTO SaveCreateStoreData (DataID, UID, playTime, creatingDates,
Cname)
VALUES
    (DT00000304, U000000301, '2024-02-01', 'player1'),
    (DT00000305, U000000302, '2024-02-01', 'player2'),
    (DT00000306, U000000303, '2024-02-01', 'player3'),
    (DT00000307, U000000304, '2024-02-01', 'player4'),
    (DT00000308, U000000305, '2024-02-01', 'player5'),
    (DT00000309, U000000306, '2024-02-01', 'player6');

INSERT INTO Characters_Stats (HP, Level)
VALUES
    (100, 1),
    (200, 2),
    (300, 3),
    (400, 4),
    (500, 5),
    (600, 6);

INSERT INTO Characters_Info (Level, Money, Cname, Rname, MapID,
currLoc)
VALUES
    (1, 1000, 'player1', 'wizard', 'M000000305', 'lecture hall'),
    (1, 1000, 'player2', 'knight', 'M000000306', 'checkpoint2'),
    (6, 10000, 'player3', 'assassin', 'M000000307', 'checkpoint2'),
    (1, 1000, 'player4', 'knight', 'M000000308', 'checkpoint2'),
    (1000, 100000, 'player6', 'knight', 'M000000309', 'checkpoint10');
```

```
INSERT INTO Weapons (WeaponID, wpDamage, Price, Rname)
VALUES
    (WP00000304, 100000, 100000, 'wizard'),
    (WP00000306, 100000, 10000, 'knight'),
    (WP00000307, 10000, 10000, 'doctor'),
    (WP00000308, 100, 1000, 'apprentice'),
    (WP00000309, 10000000, 10000000, 'assassin');

INSERT INTO Roles (Cname, Rname)
VALUES
    ('player1', 'wizard'),
    ('player2', 'knight'),
    ('player3', 'doctor'),
    ('player4', 'apprentice'),
    ('player5', 'assassin');

INSERT
INTO LockedArea(MapID, MapName, FoggyArea)
VALUES
('M000000304', “map1”, “area1”),
('M000000305', “map2”, “area2”),
('M000000306', “map3”, “area3”),
('M000000307', “map4”, “area4”),
('M000000308', “map5”, “area5”)
```

```
INSERT
INTO UnlockArea(MapID, MapName, CheckPoint)
VALUES
('M000000304', “map6”, “checkpoint6”),
('M000000305', “map10”, “checkpoint10”),
('M000000306', “map12”, “checkpoint12”),
('M000000307', “map15”, “checkpoint15”),
('M000000308', “map17”, “checkpoint17”)

INSERT
INTO Coordinate (X_Coord, Y_Coord, MapID)
VALUES
    (X000000001, Y000000001, 'M000000304'),
    (X000000011, Y000000101, 'M000000305'),
    (X000000021, Y000000201, 'M000000306'),
    (X000000031, Y000000301, 'M000000307'),
    (X000000041, Y000000401, 'M000000308');

INSERT
INTO LearnSkills_Stats(Sname, SDamage, Requirement)
VALUES
(“invisibility”, 0,  “level 1”),
(“fireball”, 100,  “level 2”),
(“iceball”, 100,  “level 3”),
(“fly”, 0,  “level 4”),
(“speed increase”, 0,  “level 5”),
(“duplicate character”, 1000,  “level 6”),
INSERT
INTO LearnSkills_Info(skillID, Sname, IsLearned, Rname)
VALUES
(SK00000001,“invisibility”, false, “knight” ),
(SK00000002,“fireball”, false,  “apprentice”),
(SK00000003,“iceball”, false,  “wizard”),
(SK00000004,“fly”, false,  “assassin”),
(SK00000005,“speed increase”, false,  “doctor”),

INSERT
INTO BossInfo (Bname, Blevel, BossDMG, BossHP)
VALUES
        (‘BOSS1’, 1, 100, 100),
        (‘BOSS2’, 2, 200, 200),
        (‘BOSS3’, 3, 300, 300),
        (‘BOSS4’, 4, 400, 400),
        (‘BOSS9’, 99, 900, 900);

INSERT
INTO DungeonStats (dungeonName, clearStatus)
VALUES
        (‘Dungeon1’, true),
        (‘Dungeon2’, true),
        (‘Dungeon3’, false),
        (‘Dungeon4’, true),
        (‘Dungeon5’, false);
```

```
INSERT
INTO DungeonInfo (dungeonID, dungeonName, Item, Bname, BossID,
MapID)
VALUES
	(‘D000000001’, ‘Dungeon1’, ‘Item1’, ‘BOSS1’, ‘B000000001’,
'M000000304'),
	(‘D000000002’, ‘Dungeon2’, ‘ Item2’, ‘BOSS2’, ‘B000000002’,
'M000000305'),
	(‘D000000003’, ‘Dungeon3’, ‘Item3’, ‘BOSS3’, ‘B000000003’,
'M000000306'),
	(‘D000000004’, ‘Dungeon4, ‘Item4’, ‘BOSS4’, ‘B000000004’,
'M000000307'),
	(‘D000000005’, ‘Dungeon5, ‘Item5’, ‘BOSS5’, ‘B000000005’,
'M000000308');
```