University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 2025.03.03

Group Number: 90

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alexi Manning	58046186	l0g8p	aleximanning7@gmail.com
Livia Lin	44851947	k1m7a	livialin2014@gmail.com
Yoobeen Hong	51765535	m2v3m	yoobeenhong@gmail.com

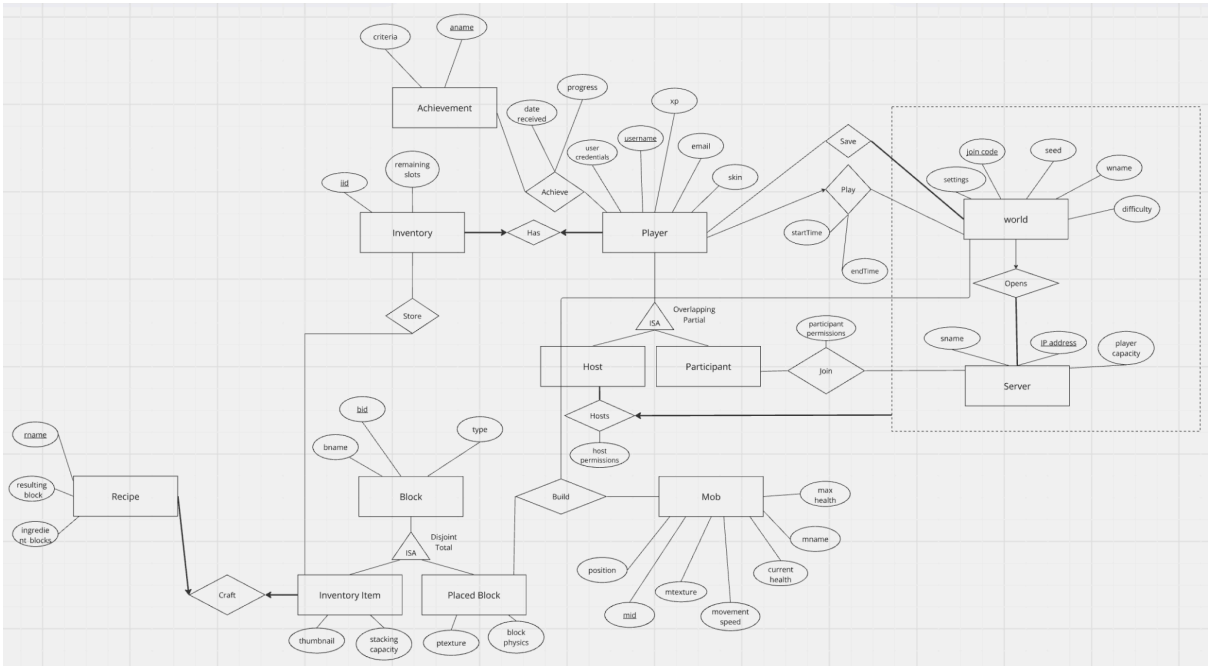
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

Brief Summary of Project

This database models the elements of the game Minecraft, including blocks, players, achievements, worlds, and servers. It models how the quantity of different types of objects like blocks, recipes, and mobs may change over time as a player plays in a world. It also represents how a world itself may also be part of a larger server, hosted by certain players and enjoyed by other players.

ER Diagram



Changes from Milestone 1:

1. Placed Block is now part of the *Build* relationship instead of Block. The Block ISA relationship has total covering, so a Block entity will always either be an Inventory Item or a Placed Block. Inventory Item entities will never be “built,” so only Placed Blocks should be part of the *Build* relationship instead of all Blocks.
2. Inventory Item is now part of the *Store* relationship instead of Block. The Block ISA relationship has total covering, so a Block entity will always either be an Inventory Item or a Placed Block. Placed Blocks will never be “stored,” so only Inventory Items should be part of the *Store* relationship instead of all Blocks.
3. The *maximum slots* attribute of the Inventory entity was removed. The maximum slots an Inventory has will be the same for every Inventory entity (constant), so there is no need to store it as an attribute in the database.
4. Participation constraint added for Host in the *Hosts* relationship, since Host entities become “hosts” instead of just “players” when they host a Server with a World. If they are not hosting any servers, then they are just a Player or a Participant in another Server.
5. The *date received* and *progress* attributes of the Achievement entity were moved to be part of the *Achieve* relationship instead. Achieve is a many-to-many relationship between Achievement and Player, so many Players can “achieve” the same Achievement. Thus, these attributes should be part of the *Achieve* relationship rather than the Achievement itself so that the Players won’t have the inaccurate *date received* and *progress* attributes that were actually given to other players.

6. A *criteria* attribute was added to the Achievement entity, in order to allow the database to store a description of what tasks need to be completed in order for a Player to receive an achievement.
7. The *SinglePlay* relationship was renamed to *Play* for clarity.

Schema
Primary keys are underlined and foreign keys are **bolded**.

Inventory(iid: INTEGER, remaining_slots: INTEGER)

- remaining_slots must be NOT NULL.

Server(IPaddress: VARCHAR(15), sname: VARCHAR(255), player_capacity: INTEGER)

PlayerHas(username: VARCHAR(16), user_credentials: VARCHAR(255), xp: INTEGER, email: VARCHAR(255), skin: INTEGER, **iid:** INTEGER)

- user_credentials must be NOT NULL.
- email is a CANDIDATE KEY.
- email must be UNIQUE.
- email must be NOT NULL.
- iid must be UNIQUE.
- iid must be NOT NULL.

Host(**username**: VARCHAR(16))

- The participation constraint of Host in the Hosts relationship from the ER Diagram cannot be translated in the relational schema.

Participant(**username**: VARCHAR(16))

Join(**username:** VARCHAR(16), **IPaddress:** VARCHAR(15), participant_permissions: VARCHAR(255))

WorldOpensHosts(join_code: VARCHAR(255), wname: VARCHAR(255), seed: INTEGER, settings: VARCHAR(255), difficulty: VARCHAR(255), host_permissions: VARCHAR(255), **IPaddress:** VARCHAR(15), **username:** VARCHAR(16))

- IPaddress must be NOT NULL.
- username must be NOT NULL.

Play(**username:** VARCHAR(16), **join_code:** VARCHAR(255), start_time: TIMESTAMP(0), end_time: TIMESTAMP(0))

- Though Play is a one-to-many relationship between World and Player and should be combined into one table with Player, we separated it into its own table so that the tables including the entities World and Player don't have foreign keys that reference each other. Otherwise, our SQL DDL statements would not run.

Save(**join_code**: VARCHAR(255), **username**: VARCHAR(16))

- join_code must be NOT NULL.

InventoryItem(bid**:** INTEGER, bname: VARCHAR(255), type: VARCHAR(255), thumbnail: INTEGER, stacking_capacity: INTEGER)

PlacedBlock(bid**:** INTEGER, bname: VARCHAR(255), type: VARCHAR(255), ptexture: INTEGER, block_physics: VARCHAR(255))

Mob(mid: INTEGER, mname: VARCHAR(255), mtexture: INTEGER, position: VARCHAR(255), max_health: INTEGER, current_health: INTEGER, movement_speed: DECIMAL)

Store(**bid**: INTEGER, **iid**: INTEGER)

Build(**join_code**: VARCHAR(255), **bid:** INTEGER, **mid**: INTEGER)

RecipeCraft(rname: VARCHAR(255), **resulting_block**: VARCHAR(255), ingredient_block: VARCHAR(255))

- resulting_block is both a FOREIGN KEY that REFERENCES the bid of Inventory Item and an attribute of Recipe in the ER Diagram.
- resulting_block must be UNIQUE.
- resulting_block must be NOT NULL.

Achievement(aname: VARCHAR(255), criteria: VARCHAR(255))

Achieve(**username**: VARCHAR(16), **aname**: VARCHAR(255), date_received: DATE, progress: DECIMAL)

Functional Dependencies

PlayerHas:
username → user_credentials, xp, email, skin, iid
email → username, user_credentials, xp, skin, iid

Play:
username → join_code, start_time, end_time

Join:
username, IPaddress → participant_permissions

Server:
IPaddress → sname, player_capacity

InventoryItem:
bid → bname, type
    bname → stacking_capacity, thumbnail

PlacedBlock:
bid → bname, type
    bname → block_physics, ptexture

Mob:
mid → position, mtexture, movement_speed, current_health, mname, max_health
    mname → texture, position, max_health, current_health, movement_speed

Achievement:
aname → criteria

Achieve:
username, aname → date_received, progress

Inventory:
iid → remaining_slots

RecipeCraft:
rname → resulting_block, ingredient_blocks
    resulting_block → ingredient_blocks
    ingredient_blocks → resulting_block

WorldOpensHosts:
join_code → wname, seed, settings, difficulty, host_permissions, IPaddress, username
    seed → difficulty

Normalization

We will decompose our tables into BCNF, not 3NF.

PlayerHas is in BCNF.
Play is in BCNF.
Host is in BCNF.
Participant is in BCNF.
Play is in BCNF.
Join is in BCNF.
Server is in BCNF.

InventoryItem(<u>bid</u>, bname, type, stacking_capacity, thumbnail) is not in BCNF.

FDs:
1. bid → bname, type
2. bname → stacking_capacity, thumbnail

Closures of LHS:
- bid+ = {bid, bname, type, stacking_capacity, thumbnail}
- bname+ = {bname, stacking_capacity, thumbnail}

FD2 violates BCNF since bname+ does not include all attributes of InventoryItem.

Decomposition into BCNF:
1. bname → stacking_capacity
   InventoryItem1(<u>bname</u>, stacking_capacity), InventoryItem'(bname, bid, thumbnail, type)
2. bname → thumbnail
   InventoryItem2(<u>bname</u>, thumbnail), InventoryItem3(**bname**, <u>bid</u>, type)
3. Combine InventoryItem1 and InventoryItem2 into one relation since they both have the same key (bname).

Final relations:
InventoryItem1(<u>bname</u>, stacking_capacity, thumbnail), InventoryItem2(**bname**, <u>bid</u>, type)

PlacedBlock(<u>bid</u>, bname, type, ptexture, block_physics) is not in BCNF.
FDs:
1. bid → bname, type
2. bname → block_physics, ptexture

Closures of LHS:
- bid+ = {bid, bname, type, block_physics, ptexture}
- bname+ = {bname, block_physics, ptexture}

FD2 violates BCNF since bname+ does not include all attributes of PlacedBlock.

Decomposition into BCNF:

1. bname → block_physics
   PlacedBlock1(<u>bname</u>, block_physics), PlacedBlock'(bname, ptexture, bid, type)
2. bname → ptexture
   PlacedBlock2(<u>bname</u>, ptexture), PlacedBlock3(**bname**, <u>bid</u>, type)
3. Combine PlacedBlock1 and PlacedBlock2 into one relation since they both have the same key (bname).

Final relations:
PlacedBlock1(<u>bname</u>, block_physics, ptexture), PlacedBlock2(**bname**, <u>bid</u>, type)

Mob(<u>mid</u>, mname, mtexture, position, max_health, current_health, movement_speed) is not in BCNF.
FDs:

1. mid → position, mtexture, movement_speed, current_health, mname, max_health
2. mname → mtexture, position, max_health, current_health, movement_speed

Closures of LHS:

- mid+ = {mid, position, mtexture, movement_speed, current_health, name, max_health}
- mname+ = {mname, mtexture, position, max_health, current_health, movement_speed}

FD2 violates BCNF since mname+ does not contain all attributes of Mob.

Decomposition into BCNF:

1. mname → mtexture, position, max_health, current_health, movement_speed
   Mob1(<u>mname</u>, mtexture, position, max_health, current_health, movement_speed), Mob2(<u>mid</u>, **mname**)

Final relations:
Mob1(<u>mname</u>, mtexture, position, max_health, current_health, movement_speed), Mob2(<u>mid</u>, **mname**)

Achievement is in BCNF.
Achieve is in BCNF.
Inventory is in BCNF.
Save is in BCNF.
Build is in BCNF.
Store is in BCNF.

RecipeCraft(<u>rname</u>, **resulting_block**, ingredient_blocks) is not in BCNF.
FDs:

1. rname → resulting_block, ingredient_blocks

2. resulting_block → ingredient_blocks
3. ingredient_blocks → resulting_block

Closures of LHS:
- rname+ = {rname, resulting_block, ingredient_blocks}
- resulting_block+ = {resulting_block, ingredient_blocks}
- Ingredient_blocks+ = {ingredient_blocks, resulting_block}

FD2 violates BCNF since resulting_block+ does not contain all attributes of RecipeCraft.

Decomposition into BCNF:
1. resulting_block → ingredient_blocks
   RecipeCraft1(**resulting_block**, ingredient_blocks), RecipeCraft2(rname, **resulting_block**)

Final relations:
RecipeCraft1(**resulting_block**, ingredient_blocks), RecipeCraft2(rname, **resulting_block**)

WorldOpensHosts(join_code, wname, seed, settings, difficulty, **IPaddress**, host_permissions, **username**) is not in BCNF.
FDs:
1. join_code → wname, seed, settings, difficulty, host_permissions, IPaddress, username
2. seed → difficulty

Closures of LHS:
- join_code+ = {join_code, wname, seed, settings, difficulty, IPaddress, host_permissions, username}
- seed+ = {seed, difficulty}

FD2 violates BCNF since seed+ does not include all attributes of WorldOpensHosts.

Decomposition into BCNF:
1. seed → difficulty
   WorldOpensHosts1(seed, difficulty), WorldOpensHosts2(join_code, wname, **seed**, settings, **IPaddress**, host_permissions, **username**)

Final relations:
WorldOpensHosts1(seed, difficulty)
WorldOpensHosts2(join_code, wname, **seed**, settings, **IPaddress**, host_permissions, **username**)

Final Tables:

Inventory(iid: INTEGER, remaining_slots: INTEGER)

Server(IPaddress: VARCHAR(15), sname: VARCHAR(255), player_capacity: INTEGER)

PlayerHas(username: VARCHAR(16), user_credentials: VARCHAR(255), xp: INTEGER, email: VARCHAR(255), skin: INTEGER, **iid:** INTEGER)

- email is a CANDIDATE KEY.

Host(**username**: VARCHAR(16))

Participant(**username**: VARCHAR(16))

Join(**username:** VARCHAR(16), participant_permissions: VARCHAR(255), **IPaddress:** VARCHAR(15))

WorldOpensHosts1(seed: INTEGER, difficulty: VARCHAR(255))

WorldOpensHosts2(join_code: VARCHAR(255), wname: VARCHAR(255), **seed:** INTEGER, Settings: VARCHAR(255), **IPaddress:** VARCHAR(15), host_permissions: VARCHAR(255), **username:** VARCHAR(16))

Play(**username:** VARCHAR(255), **join_code:** VARCHAR(255), start_time: TIMESTAMP(0), end_time: TIMESTAMP(0))

Save(**join_code**: VARCHAR(255), **username**: VARCHAR(16))

InventoryItem1(bname: VARCHAR(255), stacking_capacity: INTEGER, thumbnail: INTEGER)

InventoryItem2(**bname**: VARCHAR(255), bid: INTEGER, type: VARCHAR(255))

PlacedBlock1(bname: VARCHAR(255), block_physics: VARCHAR(255), ptexture: INTEGER)

PlacedBlock2(**bname**: VARCHAR(255), bid: INTEGER, type: VARCHAR(255))

Mob1(mname: VARCHAR(255), mtexture: INTEGER, position: VARCHAR(255), max_health: INTEGER, current_health: INTEGER, movement_speed: DECIMAL)

Mob2(mid: INTEGER, **mname**: VARCHAR(255))

Store(**bid**: INTEGER, **iid**: INTEGER)

Build(**bid:** INTEGER, **join_code**: VARCHAR(255), **mid**: INTEGER)

RecipeCraft1(**resulting_block**: VARCHAR(255), ingredient_blocks: VARCHAR(255))

RecipeCraft2(rname: VARCHAR(255), **resulting_block**: VARCHAR(255))

Achievement(aname: VARCHAR(255), criteria: VARCHAR(255))

Achieve(**username**: VARCHAR(16), **aname**: VARCHAR(255), date_received: DATE, progress: DECIMAL)

SQL DDL Statements

Our SQL DDL statements were tested using Oracle, since we are planning to implement our project using Oracle.

```
CREATE TABLE Inventory(
        iid                 INTEGER     PRIMARY KEY,
        remaining_slots     INTEGER     NOT NULL
);

CREATE TABLE Server (
        IPaddress           VARCHAR(15)     PRIMARY KEY,
        sname               VARCHAR(255),
        player_capacity     INTEGER
);

CREATE TABLE PlayerHas (
        username            VARCHAR(16)     PRIMARY KEY,
        user_credentials    VARCHAR(255)    NOT NULL,
        xp                  INTEGER,
        email               VARCHAR(255)    NOT NULL    UNIQUE,
        skin                INTEGER,
        iid                 INTEGER         NOT NULL    UNIQUE,
        FOREIGN KEY (iid) REFERENCES Inventory(iid)
                ON DELETE CASCADE
);

CREATE TABLE Host (
        username    VARCHAR(16)     PRIMARY KEY,
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE CASCADE
);

CREATE TABLE Participant (
        username    VARCHAR(16)     PRIMARY KEY,
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE CASCADE
);

CREATE TABLE Join (
        username                VARCHAR(16),
        IPaddress               VARCHAR(15),
        participant_permissions VARCHAR(255),
        PRIMARY KEY (username, IPaddress),
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE CASCADE,
        FOREIGN KEY (IPaddress) REFERENCES Server(IPaddress)
                ON DELETE CASCADE
);
```

```
CREATE TABLE WorldOpensHosts1 (
        seed            INTEGER      PRIMARY KEY,
        difficulty      VARCHAR(255)
);

CREATE TABLE WorldOpensHosts2 (
        join_code               VARCHAR(255)      PRIMARY KEY,
        settings                VARCHAR(255),
        host_permissions        VARCHAR(255),
        wname                   VARCHAR(255),
        seed                    INTEGER,
        username                VARCHAR(16)       NOT NULL,
        IPaddress               VARCHAR(15)               NOT NULL,
        FOREIGN KEY (seed) REFERENCES WorldOpensHosts1(seed)
                ON DELETE CASCADE,
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE CASCADE,
        FOREIGN KEY (IPaddress) REFERENCES Server(IPaddress)
                ON DELETE CASCADE
);

CREATE TABLE Play(
        username      VARCHAR(16)       PRIMARY KEY,
        join_code     VARCHAR(255),
        start_time    TIMESTAMP(0),
        end_time      TIMESTAMP(0),
        FOREIGN KEY (username) REFERENCES PlayerHas(username),
        FOREIGN KEY (join_code) REFERENCES WorldOpensHosts2(join_code)
                ON DELETE SET NULL
);

CREATE TABLE Save (
        join_code     VARCHAR(255)      NOT NULL,
        username      VARCHAR(16),
        PRIMARY KEY (join_code, username),
        FOREIGN KEY (join_code) REFERENCES WorldOpensHosts2(join_code)
                ON DELETE CASCADE,
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE CASCADE
);

CREATE TABLE InventoryItem1 (
        bname                   VARCHAR(255)      PRIMARY KEY,
        thumbnail               INTEGER,
        stacking_capacity       INTEGER
);
```

```
CREATE TABLE InventoryItem2 (
        bid             INTEGER     PRIMARY KEY,
        bname           VARCHAR(255),
        type            VARCHAR(255),
        FOREIGN KEY (bname) REFERENCES InventoryItem1(bname)
                ON DELETE CASCADE
);

CREATE TABLE PlacedBlock1 (
        bname               VARCHAR(255)        PRIMARY KEY,
        ptexture            INTEGER,
        block_physics       VARCHAR(255)
);

CREATE TABLE PlacedBlock2 (
        bid             INTEGER     PRIMARY KEY,
        bname           VARCHAR(255),
        type            VARCHAR(255),
        FOREIGN KEY (bname) REFERENCES PlacedBlock1(bname)
                ON DELETE CASCADE
);

CREATE TABLE Mob1 (
        mname               VARCHAR(255)        PRIMARY KEY,
        mtexture            INTEGER,
        position            VARCHAR(255),
        max_health          INTEGER,
        current_health      INTEGER,
        movement_speed      DECIMAL
);

CREATE TABLE Mob2 (
        mid             INTEGER     PRIMARY KEY,
        mname           VARCHAR(255),
        FOREIGN KEY (mname) REFERENCES Mob1(mname)
                ON DELETE CASCADE
);

CREATE TABLE Store (
        bid     INTEGER,
        iid     INTEGER,
        PRIMARY KEY (bid, iid),
        FOREIGN KEY (bid) REFERENCES InventoryItem2(bid)
                ON DELETE CASCADE,
        FOREIGN KEY (iid) REFERENCES Inventory(iid)
                ON DELETE CASCADE
);
```

```
CREATE TABLE Build (
        join_code       VARCHAR(255),
        bid             INTEGER,
        mid             INTEGER,
        PRIMARY KEY (join_code, bid, mid),
        FOREIGN KEY (join_code) REFERENCES WorldOpensHosts2(join_code)
                ON DELETE CASCADE,
        FOREIGN KEY (bid) REFERENCES PlacedBlock2(bid)
                ON DELETE SET NULL,
        FOREIGN KEY (mid) REFERENCES Mob2(mid)
                ON DELETE SET NULL
);

CREATE TABLE RecipeCraft1 (
        resulting_block         VARCHAR(255)        PRIMARY KEY,
        ingredient_blocks       VARCHAR(255),
        FOREIGN KEY (resulting_block) REFERENCES InventoryItem1(bname)
                ON DELETE CASCADE
);

CREATE TABLE RecipeCraft2 (
        rname                   VARCHAR(255)        PRIMARY KEY,
        resulting_block         VARCHAR(255)        NOT NULL    UNIQUE,
        FOREIGN KEY (resulting_block) REFERENCES RecipeCraft1(resulting_block)
                ON DELETE CASCADE
);

CREATE TABLE Achievement (
        aname           VARCHAR(255)        PRIMARY KEY,
        criteria        VARCHAR(255)
);

CREATE TABLE Achieve (
        username                VARCHAR(16),
        aname                   VARCHAR(255),
        date_received           DATE,
        progress                DECIMAL,
        PRIMARY KEY (username, aname),
        FOREIGN KEY (username) REFERENCES PlayerHas(username)
                ON DELETE SET NULL,
        FOREIGN KEY (aname) REFERENCES Achievement(aname)
                ON DELETE CASCADE
);
```

INSERT Statements
Our INSERT statements were tested using Oracle, since we are planning to implement our project using Oracle.

```
INSERT INTO Inventory(iid, remaining_slots) VALUES (1, 17);
INSERT INTO Inventory(iid, remaining_slots) VALUES (2, 37);
INSERT INTO Inventory(iid, remaining_slots) VALUES (3, 0);
INSERT INTO Inventory(iid, remaining_slots) VALUES (4, 27);
INSERT INTO Inventory(iid, remaining_slots) VALUES (5, 1);
INSERT INTO Inventory(iid, remaining_slots) VALUES (6, 9);

INSERT INTO Server(IPaddress, sname, player_capacity)
VALUES      ('123.456.1.1', 'server1', 100000);
INSERT INTO Server(IPaddress, sname, player_capacity)
VALUES      ('123.654.1.1', 'server2', 65000);
INSERT INTO Server(IPaddress, sname, player_capacity)
VALUES      ('123.655.1.1', NULL, 65000);
INSERT INTO Server(IPaddress, sname, player_capacity)
VALUES      ('123.656.1.1', NULL, 75000);
INSERT INTO Server(IPaddress, sname, player_capacity)
VALUES      ('123.657.1.1', 'server', 75000);

INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('Liv', 'iloveCPSC304', 999999, 'livia@student.ubc.ca', NULL, 1);
INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('Alexi', 'iloveCPSC304', 1000000, 'alexi3@student.ubc.ca', 100, 2);
INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('Ruby', 'iloveCPSC304!', 1000010, 'ruby@student.ubc.ca', 200, 3);
INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('Someone', 'loveCPSC304!', 161, 'someone@student.ubc.ca', 177, 4);
INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('Somebody', 'loveCPSC304!', 10000000, 'yo@student.ubc.ca', NULL, 5);
INSERT INTO PlayerHas(username, user_credentials, xp, email, skin, iid)
VALUES      ('MineCraftGenius', 'genius', 15000000, 'genius@student.ubc.ca', 1000, 6);

INSERT INTO Host(username) VALUES ('Liv');
INSERT INTO Host(username) VALUES ('Alexi');
INSERT INTO Host(username) VALUES ('Ruby');
```

```
INSERT INTO Host(username) VALUES('Someone');
INSERT INTO Host(username) VALUES ('Somebody');

INSERT INTO Participant(username) VALUES ('Liv');
INSERT INTO Participant(username) VALUES ('Alexi');
INSERT INTO Participant(username) VALUES ('Ruby');
INSERT INTO Participant(username) VALUES ('Someone');
INSERT INTO Participant(username) VALUES ('MineCraftGenius');

INSERT INTO Join(username, IPaddress, participant_permissions)
VALUES ('Liv', '123.456.1.1', '1');

INSERT INTO Join(username, IPaddress, participant_permissions)
VALUES ('Alexi', '123.654.1.1', '2');
INSERT INTO Join(username, IPaddress, participant_permissions)
VALUES ('Ruby', '123.655.1.1', '3');
INSERT INTO Join(username, IPaddress, participant_permissions)
VALUES ('Someone', '123.656.1.1', '4');
INSERT INTO Join(username, IPaddress, participant_permissions)
VALUES ('MineCraftGenius', '123.657.1.1', '5');

INSERT INTO WorldOpensHosts1(seed, difficulty)
VALUES (-1106759604738884840, 'Easy');
INSERT INTO WorldOpensHosts1(seed, difficulty)
VALUES (-5584399987456711267, 'Peaceful');
INSERT INTO WorldOpensHosts1(seed, difficulty)
VALUES (-1754216045272489466, 'Normal');
INSERT INTO WorldOpensHosts1(seed, difficulty)
VALUES (5101553622029575588, 'Hard');
INSERT INTO WorldOpensHosts1(seed, difficulty)
VALUES (4025804172371830787, 'Very Hard');

INSERT INTO WorldOpensHosts2(join_code, settings, host_permissions, wname, seed,
        IPaddress, username)
VALUES ('candyy', 'Creative Mode', '1', 'Giant Pale Garden', -1106759604738884840,
        '123.456.1.1', 'Liv');
INSERT INTO WorldOpensHosts2(join_code, settings, host_permissions, wname, seed,
```

```
        IPaddress, username)
VALUES ('ballooon', 'Peaceful Mode', '2', 'Sakura Season', -5584399987456711267,
        '123.654.1.1', 'Alexi');
INSERT INTO WorldOpensHosts2(join_code, settings, host_permissions, wname, seed,
        IPaddress, username)
VALUES ('ffish', 'Exploration Mode', '3', 'Frozen Edge Of The World',
        -1754216045272489466, '123.655.1.1', 'Ruby');
INSERT INTO WorldOpensHosts2(join_code, settings, host_permissions, wname, seed,
        IPaddress, username)
VALUES ('bannaa', 'Adventure Mode', '4', 'Savanna Plateau River', 5101553622029575588,
        '123.656.1.1', 'Someone');
INSERT INTO WorldOpensHosts2(join_code, settings, host_permissions, wname, seed,
        IPaddress, username)
VALUES ('geniuss', 'Hardcore Mode', '5', 'Giant Mangrove Swamp', 4025804172371830787,
        '123.657.1.1', 'MineCraftGenius');

INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('Liv', 'candyy', TO_TIMESTAMP('2025-02-28 10:49:00', 'YYYY-MM-DD
        HH24:MI:SS'), TO_TIMESTAMP('2025-02-28 11:01:00', 'YYYY-MM-DD
        HH24:MI:SS'));

INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('Alexi', 'ballooon', TO_TIMESTAMP('2025-02-28 10:48:00', 'YYYY-MM-DD
        HH24:MI:SS'), TO_TIMESTAMP('2025-02-28 11:05:00', 'YYYY-MM-DD
        HH24:MI:SS'));
INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('Ruby', 'ffish', TO_TIMESTAMP('2025-02-28 10:51:00', 'YYYY-MM-DD
        HH24:MI:SS'), TO_TIMESTAMP('2025-02-28 11:02:01', 'YYYY-MM-DD
        HH24:MI:SS'));
INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('Someone', 'bannaa', TO_TIMESTAMP('2025-03-02 22:00:00', 'YYYY-MM-DD
        HH24:MI:SS'), TO_TIMESTAMP('2025-03-28 23:57:59', 'YYYY-MM-DD
        HH24:MI:SS'));
INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('Somebody', 'bannaa', TO_TIMESTAMP('2025-02-28 10:20:00', 'YYYY-MM-DD
        HH24:MI:SS'), TO_TIMESTAMP('2025-03-01 15:20:00', 'YYYY-MM-DD
        HH24:MI:SS'));
```

```
INSERT INTO Play(username, join_code, start_time, end_time)
VALUES ('MineCraftGenius', 'geniuss', TO_TIMESTAMP('2025-04-07 23:30:00',
        'YYYY-MM-DD HH24:MI:SS'), TO_TIMESTAMP('2025-04-08 07:59:00',
        'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Save(join_code, username) VALUES ('candyy', 'Liv');
INSERT INTO Save(join_code, username) VALUES ('ballooon', 'Alexi');
INSERT INTO Save(join_code, username) VALUES ('ffish', 'Alexi');
INSERT INTO Save(join_code, username) VALUES ('bannaa', 'Ruby');
INSERT INTO Save(join_code, username) VALUES ('geniuss', 'MineCraftGenius');

INSERT INTO InventoryItem1(bname, thumbnail, stacking_capacity)
VALUES ('Golden Apple', 260, 64);
INSERT INTO InventoryItem1(bname, thumbnail, stacking_capacity)
VALUES ('Diamond Block', 264, 64);
INSERT INTO InventoryItem1(bname, thumbnail, stacking_capacity)
VALUES ('Bow', 261, 1);
INSERT INTO InventoryItem1(bname, thumbnail, stacking_capacity)
VALUES ('Bucket', 325, 64);
INSERT INTO InventoryItem1(bname, thumbnail, stacking_capacity)
VALUES ('Bread', 295, 64);

INSERT INTO InventoryItem2(bid, bname, type)
VALUES (260, 'Golden Apple', 'Food');
INSERT INTO InventoryItem2(bid, bname, type)
VALUES (264, 'Diamond Block', 'Mineral Block');
INSERT INTO InventoryItem2(bid, bname, type)
VALUES (261, 'Bow', 'Weapons');
INSERT INTO InventoryItem2(bid, bname, type)
VALUES (325, 'Bucket', 'Utility');
INSERT INTO InventoryItem2(bid, bname, type)
VALUES (295, 'Bread', 'Food');

INSERT INTO PlacedBlock1(bname, ptexture, block_physics)
VALUES      ('Crafting Table', 58, 'Flammable');
INSERT INTO PlacedBlock1(bname, ptexture, block_physics)
VALUES      ('Anvil', 58, 'Falling');
```

```
INSERT INTO PlacedBlock1(bname, ptexture, block_physics)
VALUES      ('Chipped Anvil', 145, 'Falling');
INSERT INTO PlacedBlock1(bname, ptexture, block_physics)
VALUES      ('Clay', 82, NULL);
INSERT INTO PlacedBlock1(bname, ptexture, block_physics)
VALUES      ('Red Tulip', 1088, 'Transparent');

INSERT INTO PlacedBlock2(bid, bname, type) VALUES (145, 'Anvil', 'Utility');
INSERT INTO PlacedBlock2(bid, bname, type) VALUES (2, 'Clay', 'Build');
INSERT INTO PlacedBlock2(bid, bname, type) VALUES (3, 'Clay', 'Build');
INSERT INTO PlacedBlock2(bid, bname, type) VALUES (58, 'Crafting Table', 'Utility');
INSERT INTO PlacedBlock2(bid, bname, type) VALUES (10, 'Chipped Anvil', 'Utility');



INSERT
INTO   Mob1(mname, mtexture, position, max_health, current_health, movement_speed)
VALUES      ('Creeper', 3, '155, 200, 145', 20, 20, 35.1);
INSERT
INTO   Mob1(mname, mtexture, position, max_health, current_health, movement_speed)
VALUES      ('Ocelot', 3, '156, 201, 0', 10, 10, 15.0);
INSERT
INTO   Mob1(mname, mtexture, position, max_health, current_health, movement_speed)
VALUES      ('Enderman', 10, '8, 0, 100', 40, 10, 10.5);
INSERT
INTO   Mob1(mname, mtexture, position, max_health, current_health, movement_speed)
VALUES      ('Villager', 115, '15, 20, 45', 20, 1, 5.2);
INSERT
INTO   Mob1(mname, mtexture, position, max_health, current_health, movement_speed)
VALUES      ('Cow', NULL, '156, 201, 0', 10, 10, 2.5);

INSERT INTO Mob2(mid, mname) VALUES (92, 'Cow');
INSERT INTO Mob2(mid, mname) VALUES (93, 'Cow');
INSERT INTO Mob2(mid, mname) VALUES (120, 'Villager');
INSERT INTO Mob2(mid, mname) VALUES (58, 'Enderman');
INSERT INTO Mob2(mid, mname) VALUES (59, 'Enderman');

INSERT INTO Store(bid, iid) VALUES (260, 1);
```

```
INSERT INTO Store(bid, iid) VALUES (264, 2);
INSERT INTO Store(bid, iid) VALUES (261, 3);
INSERT INTO Store(bid, iid) VALUES (325, 5);
INSERT INTO Store(bid, iid) VALUES (295, 6);

INSERT INTO Build(join_code, bid, mid) VALUES ('candyy', 145, 92);
INSERT INTO Build(join_code, bid, mid) VALUES ('ballooon', 2, 93);
INSERT INTO Build(join_code, bid, mid) VALUES ('ffish', 3, 120);
INSERT INTO Build(join_code, bid, mid) VALUES ('bannaa', 58, 58);
INSERT INTO Build(join_code, bid, mid) VALUES ('geniuss', 10, 59);

INSERT INTO RecipeCraft1(resulting_block, ingredient_blocks)
VALUES ('Golden Apple', 'Apple, Gold Nuggetx8');
INSERT INTO RecipeCraft1(resulting_block, ingredient_blocks)
VALUES ('Diamond Block', 'Diamondx9');
INSERT INTO RecipeCraft1(resulting_block, ingredient_blocks)
VALUES ('Bow', 'Stringx3, Stickx3');
INSERT INTO RecipeCraft1(resulting_block, ingredient_blocks)
VALUES ('Bucket', 'Iron Ingotx3');
INSERT INTO RecipeCraft1(resulting_block, ingredient_blocks)
VALUES ('Bread', 'Wheatx3');

INSERT INTO RecipeCraft2(rname, resulting_block)
VALUES ('Golden Apple Recipe', 'Golden Apple');
INSERT INTO RecipeCraft2(rname, resulting_block)
VALUES ('Diamond Block Recipe', 'Diamond Block');
INSERT INTO RecipeCraft2(rname, resulting_block)
VALUES ('Bow Recipe', 'Bow');
INSERT INTO RecipeCraft2(rname, resulting_block)
VALUES ('Bucket Recipe', 'Bucket');
INSERT INTO RecipeCraft2(rname, resulting_block)
VALUES ('Bread Recipe', 'Bread');

INSERT INTO Achievement(aname, criteria)
VALUES          ('Taking Inventory', 'Open your inventory.');
INSERT INTO Achievement(aname, criteria)
VALUES         ('Getting Wood', 'Punch a tree until a block of wood pops out.');
```

```
INSERT INTO Achievement(aname, criteria)
VALUES      ('Playing Minecraft', NULL);
INSERT INTO Achievement(aname, criteria)
VALUES      ('Acquire Hardware', 'Smelt an iron ingot.');
INSERT INTO Achievement(aname, criteria)
VALUES      ('Bake Bread', 'Turn wheat into bread.');

INSERT INTO Achieve(username, aname, date_received, progress)
VALUES      ('Liv', 'Getting Wood', TO_DATE('2025-03-02', 'YYYY-MM-DD'), 100.00);
INSERT INTO Achieve(username, aname, date_received, progress)
VALUES      ('Liv', 'Playing Minecraft', TO_DATE('2025-03-02', 'YYYY-MM-DD'), NULL);
INSERT INTO Achieve(username, aname, date_received, progress)
VALUES      ('Alexi', 'Getting Wood', NULL, 0.0);
INSERT INTO Achieve(username, aname, date_received, progress)
VALUES      ('Someone', 'Bake Bread', NULL, 98.01);
INSERT INTO Achieve(username, aname, date_received, progress)
VALUES      ('Ruby', 'Acquire Hardware', TO_DATE('2025-03-02', 'YYYY-MM-DD'), 100.0);
```

AI Acknowledgement
No AI assistance used for this milestone.