Let's apply Normalization to your Habit Tracking ER Diagram step by step:

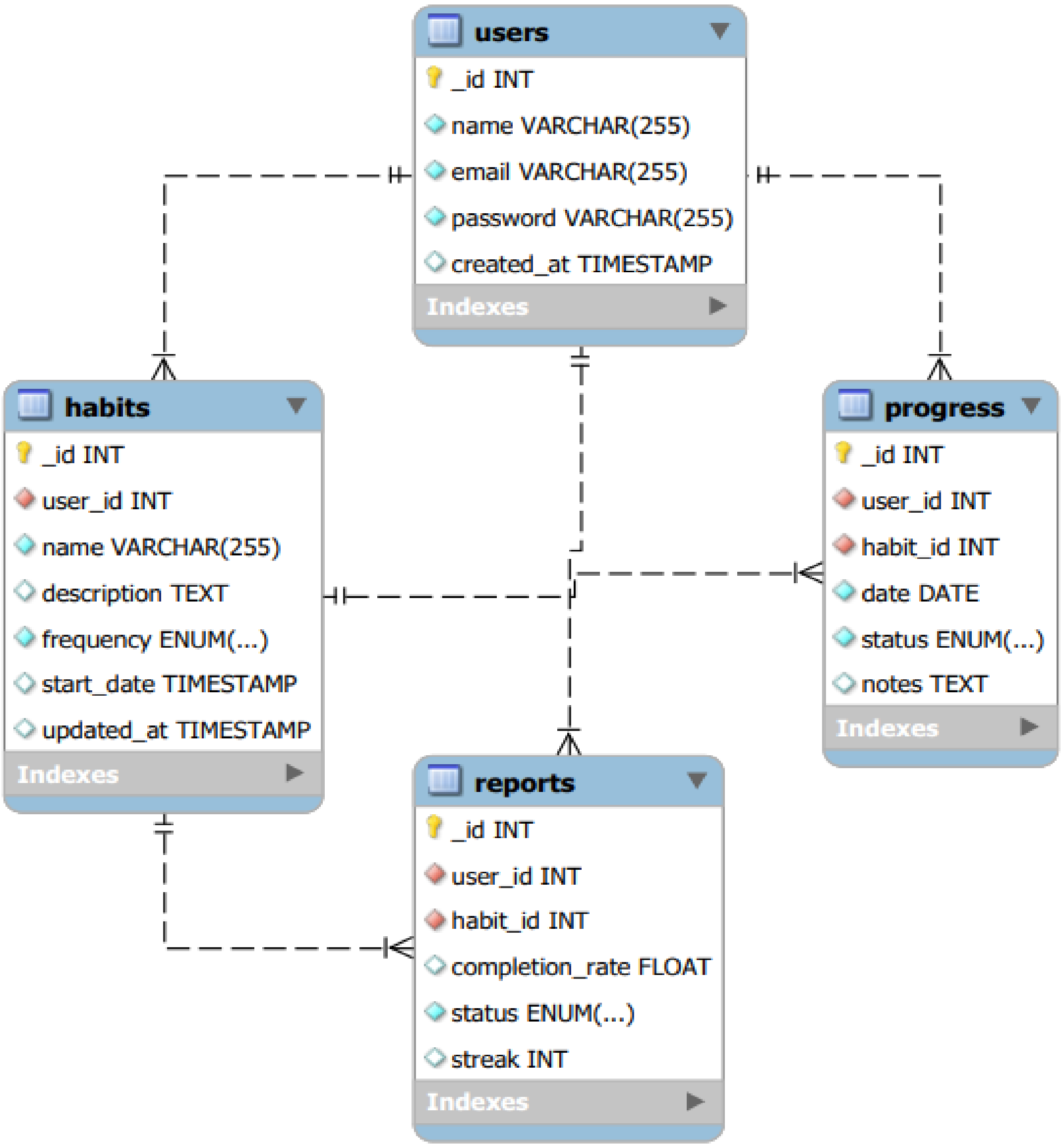

Fig: Habit Tracking ER Diagram

1st Normal Form (1NF)

A table is in 1NF if:

- It has a primary key
- All columns contain atomic values (no repeating groups or arrays)
- Each column contains a single value (no multiple values in one column)

Issues in your schema

- The frequency and status columns might store multiple values, which is not atomic.

Solution for 1NF

- Ensure frequency is stored as an ENUM or separate table if needed.
- Ensure status is atomic.

No major redesign is needed, assuming ENUM values are atomic.

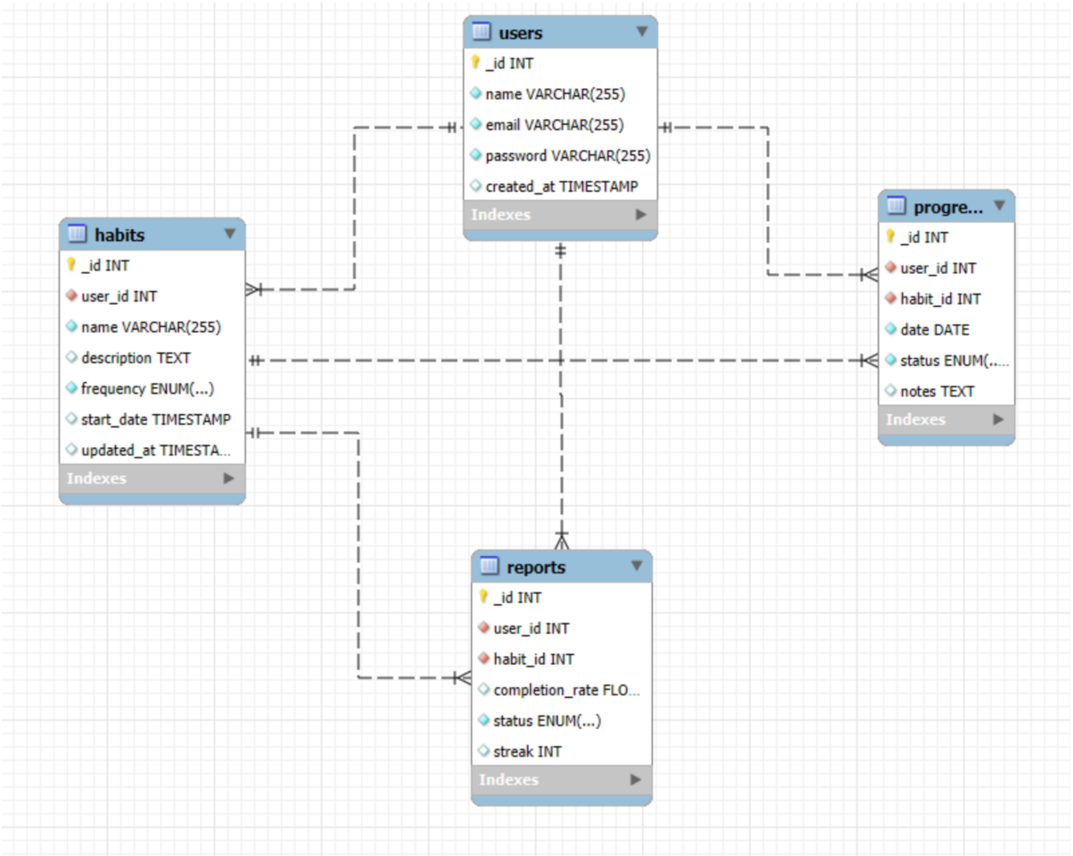

Fig: First Normal Form

Changes Applied for 1NF

✓ Atomic values ensured in frequency and status using ENUM.

✓ Each field stores only a single value (no lists or arrays).

✓ Primary keys added for uniqueness.

✓ Indexes created for faster queries.

2nd Normal Form (2NF)

A table is in 2NF if:

- It is already in 1NF
- It has no partial dependencies (No non-key attribute should depend on a part of a composite key)

Issues in your schema

- The progress table has (habit_id, user_id, date) but notes is dependent only on habit_id.
- The reports table has (habit_id, user_id) but completion_rate, streak, and status are only dependent on habit_id.

Solution for 2NF

- Move notes from progress to a separate habit_notes table.
- Move completion_rate, streak, and status from reports to a separate habit_statistics table.

Updated Tables

1. habit_notes
   - _id (INT) - Primary Key
   - habit_id (INT) - Foreign Key
   - note (TEXT)
   - date (DATE)
2. habit_statistics
   - _id (INT) - Primary Key
   - habit_id (INT) - Foreign Key
   - completion_rate (FLOAT)
   - streak (INT)
   - status (ENUM(...))

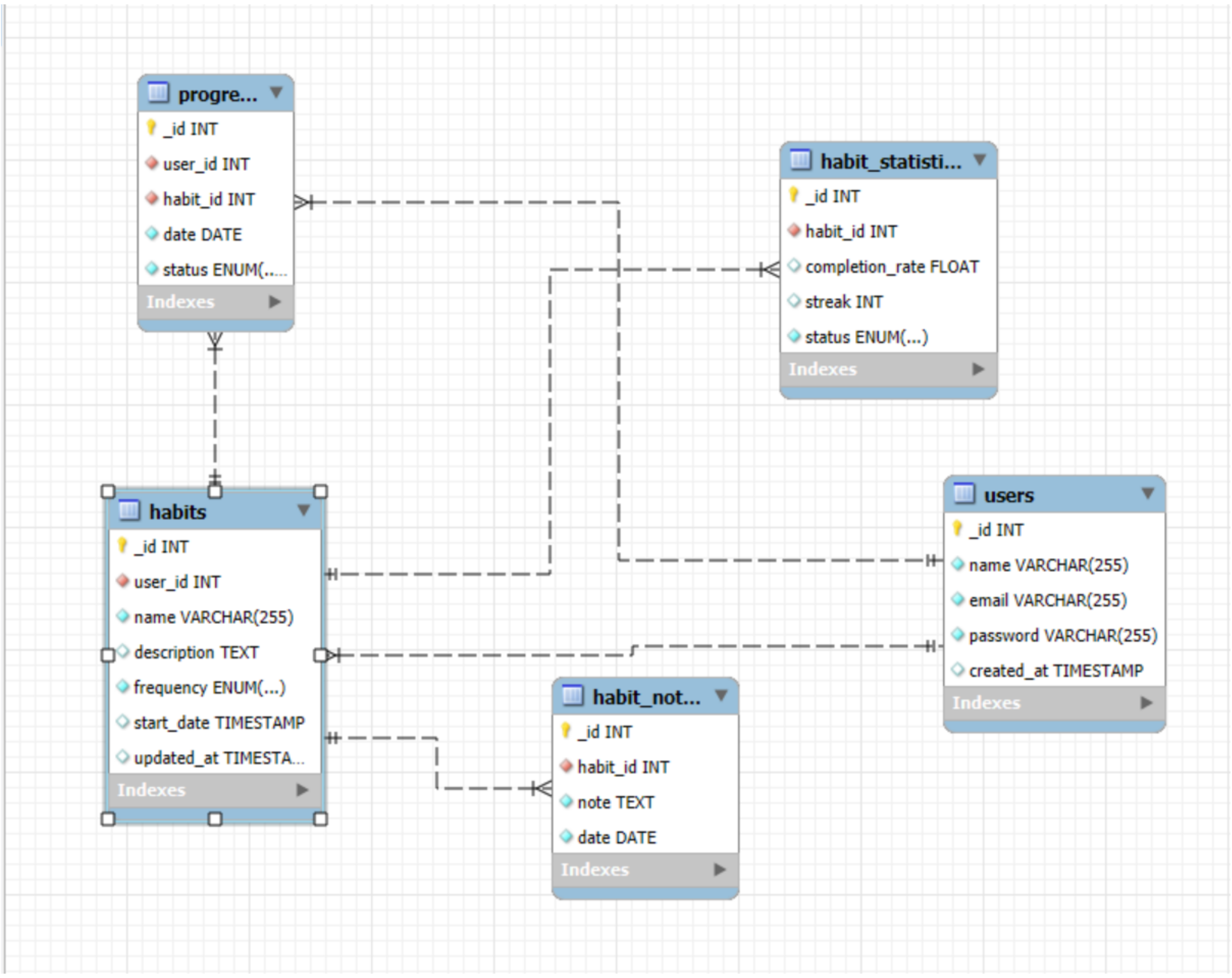

Fig: Second Normal Form

Changes Applied for 2NF:

✅ Removed Partial Dependencies by creating separate tables:

- habit_notes (for storing habit-related notes separately).
- habit_statistics (for storing completion rate, streak, and status separately).

  ✅ No non-key column depends on part of a composite key.

Improvements in 2NF

✅ No partial dependencies – all non-key columns are fully dependent on the primary key.

✅ Separated non-dependent attributes – habit_notes and habit_statistics ensure data integrity.

✅ Better structure for reporting & tracking progress.

3rd Normal Form (3NF)

A table is in 3NF if:

- It is in 2NF
- It has no transitive dependencies (A non-key column should not depend on another non-key column)

Issues in your schema

- The users table has email and password. If email is unique, it can act as a candidate key.
- In reports, completion_rate influences status (e.g., if completion rate = 100%, status = "Completed").
- The progress table has status, which may depend on habit_id and not just the primary key.

Solution for 3NF

- Split the users table into user_auth and user_profile
- Remove status from progress if it is derived from habit_statistics

Updated Tables

1. user_profile
   - _id (INT) - Primary Key
   - name (VARCHAR(255))
   - created_at (TIMESTAMP)
2. user_auth
   - _id (INT) - Primary Key
   - email (VARCHAR(255)) - Unique
   - password (VARCHAR(255))

Changes Applied for 3NF:

✅ Removed Transitive Dependencies

- Separated user_profile from user_auth to ensure email and password are independent.
- Removed derived values (status in progress, as it depends on habit_statistics).

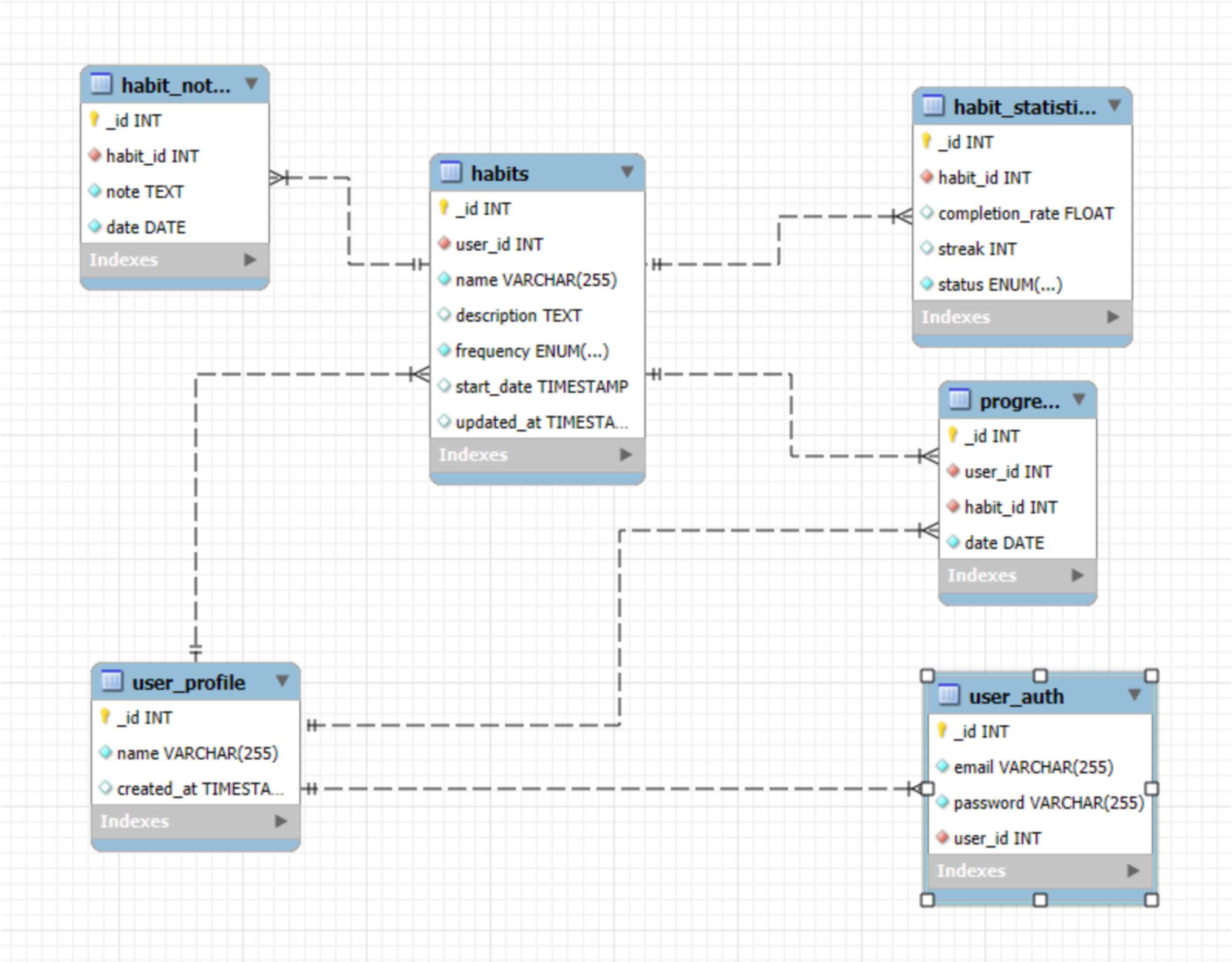

Fig: Third Normal Form

Improvements in 3NF

✅ No transitive dependencies – user_auth and user_profile separate login data from personal info.

✅ Derived values removed – progress.status is now handled by habit_statistics.

✅ Clearer, optimized structure – better organization for authentication and habit tracking.