

CUB Sales – ER to SQL Mapping Report

Yassin Soliman, Haitham Essahli, Fatima Fares

Introduction

This report explains how our ER diagram for **CUB Sales** was turned into an SQL schema. The ER diagram shows the main parts of the system and how they connect, while the SQL schema makes these ideas work in a real database. Our goal was to keep the design clear, correct, and easy to extend.

Entities to Tables

Every main entity in the ER diagram was mapped to a table:

- **User** → `user`, with `netid`, `email`, and `display_name`. Both `netid` and `email` must be unique.
- **Listing** → `listing`, linked to the posting user (`poster_id`) and storing details such as `title`, `description`, `price`, and `status`.
- **Category** → `category`, used to group listings. Each category has a unique name.
- **Contact** → `contact`, which stores messages from users about listings.

Relationships to Tables

Relationships were mapped in standard ways:

- **User–Listing (1:N)**: stored with a foreign key (`poster_id`) in `listing`.
- **Listing–Category (M:N)**: implemented with a junction table `listing_category`, using a composite primary key.
- **User–Listing (Favorite, M:N)**: stored in the `favorite` table so users can save posts they like.
- **Contact–Listing (1:N)**: each contact is linked to a listing by a foreign key.

ISA Hierarchies

Our ER diagram has three ISA hierarchies. We used the **class-table inheritance** approach (one table per subclass). This avoids NULL values and keeps data clean.

- **User ISA**: `admin_user` and `student_user`, each linked to `user`.
- **Listing ISA**: `sales_listing` and `housing_listing`, each linked to `listing`.
- **Contact ISA**: `email_contact` and `phone_contact`, each linked to `contact`.

For example, housing listings include fields like `rooms` and `monthly_rent`, which do not apply to sales listings. Keeping subclasses separate makes the schema easier to understand and manage.

Constraints

We used several SQL constraints to keep the data valid:

- **PRIMARY KEY** on all tables.
- **FOREIGN KEY** to enforce links between tables.
- **NOT NULL** for required values like **email** or **title**.
- **UNIQUE** for values that cannot repeat, such as **netid** or **category name**.
- **ENUM** for fixed sets of values, like **status** in listings or **method** in contacts.

Alternatives Considered

Another option was **single-table inheritance**, where all subclasses are stored in one table with a **type** column. We decided against this because it would cause many **NULL** fields and mix unrelated data. Separate tables gave us a clearer and safer design.

Conclusion

The mapping from ER to SQL was done to stay close to the design while enforcing strong rules in MySQL. Entities became main tables, relationships were added through foreign keys or junction tables, and ISA hierarchies were handled with class-table inheritance. The result is a schema that is simple, consistent, and ready for future use in CUB Sales.